



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**COMPARISON OF A NOISE-VARIANCE WEIGHTED
COMPLEX EXPONENTIAL RECONSTRUCTOR WITH
TRADITIONAL RECONSTRUCTORS IN THE PRESENCE
OF DEEP TRUBULENCE**

by

Shane C. Moran

June 2013

Thesis Advisor:
Thesis Co-Advisor:

Brij Agrawal
Jae-Jun Kim

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE June 2013	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE COMPARISON OF A NOISE-VARIANCE WEIGHTED COMPLEX EXPONENTIAL RECONSTRUCTOR WITH TRADITIONAL RECONSTRUCTORS IN THE PRESENCE OF DEEP TRUBULENCE			5. FUNDING NUMBERS	
6. AUTHOR(S) Shane C. Moran				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) <p>The military, and specifically the Navy, is making a move toward the use of directed energy weapons as a tactical advantage over typical kinetic weapons. Directed energy weapons require different controls and delivery methods to deal with differences in interaction of the atmosphere and the laser, as compared to traditional kinetic weapons. This thesis looks at a Noise Variance Weighted Complex Exponential Reconstruction method, and compares with typical zonal and modal least squares methods, to determine the actual wavefront in the presence of disturbances characteristic of the "deep turbulence" experienced in the maritime environment. The ability of each reconstructor to handle the effects of intensity dropout, branch points, and branch cuts is analyzed along with the effects of signal strength of the sensor, sensor grid size, and level of the intensity experienced.</p>				
14. SUBJECT TERMS Noise-Variance Weighted Complex Exponential Reconstructor, Adaptive Optics, Branch Points, Deep Turbulence, Branch Cuts			15. NUMBER OF PAGES 105	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**COMPARISON OF A NOISE-VARIANCE WEIGHTED COMPLEX
EXPONENTIAL RECONSTRUCTOR WITH TRADITIONAL
RECONSTRUCTORS IN THE PRESENCE OF DEEP TRUBULENCE**

Shane C. Moran
Ensign, United States Navy
B.S., United States Naval Academy, 2012

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
June 2013**

Author: Shane C. Moran

Approved by: Dr. Brij Agrawal
Thesis Advisor

Dr. Jae-Jun Kim
Thesis Co-Advisor

Dr. Knox Millsaps
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The military, and specifically the Navy, is making a move toward the use of directed energy weapons as a tactical advantage over typical kinetic weapons. Directed energy weapons require different controls and delivery methods to deal with differences in interaction of the atmosphere and the laser, as compared to traditional kinetic weapons.

This thesis looks at a Noise Variance Weighted Complex Exponential Reconstruction method, and compares with typical zonal and modal least squares methods, to determine the actual wavefront in the presence of disturbances characteristic of the “deep turbulence” experienced in the maritime environment. The ability of each reconstructor to handle the effects of intensity dropout, branch points, and branch cuts is analyzed along with the effects of signal strength of the sensor, sensor grid size, and level of the intensity experienced.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	MOTIVATION AND BACKGROUND	1
B.	THESIS OBJECTIVE	2
C.	THESIS OVERVIEW	2
II.	ATMOSPHERIC TURBULENCE.....	5
A.	PARAMETERS AND MEASUREMENTS	5
B.	MARITIME ENVIRONMENT AND DEEP TURBULENCE.....	8
C.	BRANCH POINTS AND BRANCH CUTS.....	8
III.	ADAPTIVE OPTICS.....	11
A.	TYPICAL ADAPTIVE OPTICS SYSTEMS	11
B.	BASIC ADAPTIVE OPTICS CONTROL CONCEPTS	12
IV.	WAVEFRONT ESTIMATION	15
A.	WAVEFRONT SENSOR.....	15
B.	SHWFS OUTPUT SIMULATION METHOD	17
C.	ESTIMATION (RECONSTRUCTION) METHODS	19
1.	Zonal Reconstruction.....	20
2.	Modal Reconstruction	21
3.	Noise-Variance Weighted Complex Exponential Reconstructor ..	26
a.	<i>Reduce Stage</i>	28
b.	<i>Solve Stage.....</i>	31
c.	<i>Rebuild Stage.....</i>	33
d.	<i>Phase Unwrapping and Branch Point Analysis</i>	38
e.	<i>Adaptation to Fried Geometry</i>	41
V.	ANALYSIS	45
A.	SOURCE PHASE AND INTENSITY DATA.....	45
B.	INITIAL TEST	48
1.	Variation in Number of Sensors	48
2.	Variation in SNR.....	54
C.	EVALUATION OF RECONSTRUCTORS	60
1.	Rytov Variance of 0.109	61
2.	Rytov Variance of 0.349	62
3.	Rytov Variance of 0.567	63
D.	FINAL RESULTS.....	64
VI.	CONCLUSION	65
	APPENDIX.....	67
A.	INITIALIZATION AND SIMULATION CODE	67
B.	SHACK-HARTMANN WAVEFRONT SENSOR SIMULATION CODE	68
C.	ZONAL RECONSTRUCTION	70
D.	MODAL RECONSTRUCTION.....	71

E. NVWCER	74
LIST OF REFERENCES.....	85
INITIAL DISTRIBUTION LIST	87

LIST OF FIGURES

Figure 1.	Phase calculation paths leading to different phase values (From Pellizzari 2010)	9
Figure 2.	Typical AO System Setup (From Allen 2007)	12
Figure 3.	Classical AO Control System	13
Figure 4.	SHWFS Schematic (From Mann 2012).....	15
Figure 5.	SHWFS Slope Simulate Test Phase Screen ($z=x^2+y^3$)	18
Figure 6.	SHWFS Output Simulation for X and Y slopes based on Fig. 5 Phase.....	19
Figure 7.	Common Reconstruction Geometries (a) Southwell (b) Hudgin (c) Fried (From Southwell 1980).....	20
Figure 8.	SHWFS Grid Circumscribed inside Unit Disk for Zernike Polynomial Evaluation	25
Figure 9.	Lattice Points for Reduce Stage of NVWCER (a) N=3 (b) N=2 (From Fried 2001).....	28
Figure 10.	Calculation of Differential Phasors for Reduce stage, grid points located on edge of lattice (From Fried 2001)	29
Figure 11.	Calculation of Differential Phasors for Reduce stage, grid points located on interior of lattice (From Fried 2001).....	30
Figure 12.	Grid Patter for Solve Stage of NVWCER.....	32
Figure 13.	Calculation for center phasor of each elemental square in Rebuild stage (From Fried 2001).....	34
Figure 14.	Calculation for Phasors at the middle of edges of elementary squares, located on interior of lattice (From Fried 2001)	36
Figure 15.	Phasor calculation setup for Rebuild stage; phasor located on edge of lattice (From Fried 2001).....	37
Figure 16.	Fried-to-Hudgin Geometry Conversion (From Fried 2001).....	41
Figure 17.	Corresponding Hudgin Geometries for a SHFWS Fried Geometry (From Fried 2001).....	42
Figure 18.	Simulation data for Rytov Variance of 0.109	46
Figure 19.	Simulation data for Rytov Variance of 0.349	47
Figure 20.	Simulation data for Rytov Variance of 0.567	47
Figure 21.	NVWCER Reconstruction. S=0.146, SNR=60, and N=16.....	48
Figure 22.	Modal Reconstruction. S=0.068, SNR=60, and N=16	49
Figure 23.	Zonal Reconstruction. S=0.126, SNR=60, and N=16.....	49
Figure 24.	NVWCER Reconstruction. S=0.312, SNR=60, N=32	50
Figure 25.	Modal reconstruction. S=0.114, SNR=60, and N=32.....	51
Figure 26.	Zonal Reconstruction. S=0.238, SNR=60, and N=32.....	51
Figure 27.	NVWCER Reconstruction. S=0.297, SNR=60, and N=64.....	52
Figure 28.	Modal Reconstruction. S=0.116, SNR=60, and N=64	53
Figure 29.	Zonal Reconstruction. S=0.206, SNR=60, and N=64.....	53
Figure 30.	NVWCER Reconstructor. S=0.144, SNR=10, N=32	55
Figure 31.	Modal Reconstruction. S=0.169, SNR=10, and N=32	55
Figure 32.	Zonal Reconstruction. S=0.296, SNR=10, and N=32.....	56

Figure 33.	NVWER. $S=0.737$, $SNR=60$, and $N=32$	57
Figure 34.	Modal Reconstruction. $S=0.356$, $SNR=60$, and $N=32$	57
Figure 35.	Zonal Reconstruction. $S=0.544$, $SNR=60$, and $N=32$	58
Figure 36.	NVWCER. $S=0.763$, $SNR=200$, and $N=32$	59
Figure 37.	Modal Reconstruction. $S=0.354$, $SNR=200$, and $N=32$	59
Figure 38.	Zonal Reconstruction. $S=0.536$, $SNR=200$, and $N=32$	60
Figure 39.	Reconstructor Comparison for Rytov of 0.109. $SNR=60$ and $N=32$	61
Figure 40.	Reconstructor Comparison for Rytov of 0.349. $SNR=60$ and $N=32$	62
Figure 41.	Reconstructor Comparison for Rytov of 0.567. $SNR=60$ and $N=32$	63

LIST OF TABLES

Table 1.	First 21 Zernike Polynomial Modes (From Wyant 2003)	23
Table 2.	First 21 Zernike Polynomials in Cartesian Coordinates (From Allen 2007) ...	24
Table 3.	Results for all Reconstruction Simulations.....	64

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

SHWFS	Shack Hartmann Wave Front Sensor
ONR	Office of Naval Research
NVWCER	Noise-Variance Weighted Complex Exponential Reconstructor
DE	Directed Energy
AO	Adaptive Optics
ICBM	Inter-Continental Ballistic Missile
LaWS	Laser Weapon System
NRL	Naval Research Labs
NSLOT	Navy Surface Layer Optical Turbulence
SNR	Signal to Noise Ratio

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express my gratitude for Dr. David Fried, who mentored and assisted me throughout the learning, design, and implementation of his Noise-Variance Weighted Complex Exponential Reconstructor. Without his help much of this process would not have moved at the pace and accuracy with which it did.

I would like to thank Distinguished Professor Brij Agrawal, and Dr. Jae Jun Kim, who allowed me to operate, learn, and test in their laboratory, while providing me with priceless mentoring and guidance throughout my entire time at NPS..

I would like to thank Bautista Fernandez, whose help in generation of the simulation data and helping in understanding of zonal and modal reconstructors was paramount in my research.

I would like to thank Travis Axtell, who not only showed great interest in my research, but also provided endless knowledge and resources during my studies of wavefront reconstruction.

Each and every person provided priceless guidance and information to me during this research, and it would not have been as successful without the contributions of all of them

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MOTIVATION AND BACKGROUND

Future technologies are beginning to use both directed energy and highly accurate imaging for more advanced and efficient means of accomplishing tasks, which previously were thought to be impossible. The use of high-power, highly accurate optics has evolved with both space-based and ground-based imaging systems, allowing for exploration and discovery into areas that were previously untouched. The use of laser energy is evolving to allow for accurate and low collateral weapons, in addition to high-bandwidth communications in free-space optical communications. The initial use of these laser systems was subjected to the environment of space, which had its own non-atmospheric disturbances and issues. Now, that technology is being adapted to ground-based systems which experience not only many of the same issues space-based encountered, but also the new difficulties involved with energy and optical propagation through an environment. The possibilities of these technologies has exploded an interest amongst scientists and engineers into advancing current beam control and high-precision optical capabilities.

In recent years the Office of Naval Research (ONR) has undertaken the challenges posed by directed energy (DE) weapons with the creation of the Directed Energy Weapon Program. This program is developing a directed energy laser weapon that will increase the Navy's effectiveness at shooting down enemy weapons and/or hostile craft while minimizing collateral damage. The Navy has identified the key challenges of a DE weapon as platform stabilization, imaging and correction of the beam due to deep atmospheric turbulence, and simplifying the beam control system operational cost and complexity. Beam control for directed energy weapons, like any weapon, is necessary to correctly aim and fire the weapon in a combat environment. A primary difference between conventional weapons and a directed energy weapon is the necessity to maintain a highly precise and power-dense beam, to maximize efficiency and lower dwell time. The longer the beam travels through the atmosphere, the more the beam is absorbed and scattered. The government and civilian sectors have also started to take on

the problems involved in high-precision imaging and optical-beam control. The problems are extremely evident in the programs such as Hubble and James-Webb space telescopes.

Precision optical beam control systems can be broken down into two primary areas of control and correction; jitter control and adaptive optics. Jitter control, primarily a concern with directed energy and laser communications, focuses on the beam deviation induced by mechanical and atmospheric vibrations and inaccuracies. Adaptive optics was originally used by ground telescopes for correcting image distortion due to the atmosphere. It is also used for correction of high energy laser beam aberrations due to the atmosphere, like in the Airborne Laser (ABL). Adaptive optics is also considered to correct beam aberrations due to imperfect optical surfaces.

B. THESIS OBJECTIVE

The focus of this research is to investigate new methods for wavefront estimation and to compare with other commonly used methods. Methods and tested are entirely implemented in a simulation based environment. The research focuses on the ability of these different wavefront estimation methods to deal with the specific issues associated with a maritime environment. These specific issues include intensity dropouts, branch points, and branch cuts. The images from each reconstructor are compared to an under sampled version of the original, and an intensity-weighted average for the overall accuracy of the reconstruction is calculated. This performance measurement, commonly referred to as a Strehl ratio, is used as the primary source of comparison and evaluation of each reconstructor. Information gained from this research, will allow for confirmation on whether more advanced wavefront estimation methods are necessary for handling the problems associated with this new operating environment.

C. THESIS OVERVIEW

Chapter II provides background information and the basic goals of an adaptive optics system, as well as the basic control topology associated there in.

Chapter III provides more information on the issues concerned with DE and optical imaging propagation through the atmosphere. The specific problems created in a

maritime environment are introduced the principal concepts for adaptive optics correction is formulated.

Chapter IV introduces the concepts and importance associated with wavefront estimation, and its role in the adaptive optics control process. The primary means of wavefront sensing is introduced and modeled. Then the three wavefront reconstruction and estimation methods are introduced and explained including the two commonly used in practices today, zonal and modal estimation, and the new Noise Variance Weighted Complex Exponential (NVWCER) reconstructor.

Chapter V provides simulated results, analysis, and comparison of all three reconstruction methods. Their ability to handle the specific issues that lay within a maritime environment is analyzed.

Chapter VI provides an overall summary, conclusion, and recommendations for possible future research.

THIS PAGE INTENTIONALLY LEFT BLANK

II. ATMOSPHERIC TURBULENCE

The first step in solving a problem is fully understanding the problem itself. This section presents some background information and some commonly used parameters used to describe the effects of turbulence. There are two primary causes for turbulence in the atmosphere as a laser propagates through. The primary source is through the heating and cooling of the surface of the earth, which in turn heats and cools the air which changes its index of refraction. These differences in atmospheric temperature and density, producing differing indexes of refraction, cause the beam to diffract disproportionately across the wavefront. A secondary, and less common source, is the actual heating of the air by the laser itself, in turn also changing the index of refraction. Changes in the index of refraction cause the beam to diffract and dissipate as it propagates to the target, ultimately creating a less accurate and less power-dense beam. AO systems focus on correcting the disturbances caused by the first source, and in order to control the beam it is necessary to measure the beam profile, or the effects of the turbulence on the beam.

A. PARAMETERS AND MEASUREMENTS

There are two main paths for propagation of a DE beam: vertical, having a very high slant angle, and horizontal, having a very low slant angle. Atmospheric turbulence for vertical paths, i.e., ground-to-space architectures, has been modeled and simulated using Kolmogorov turbulence theory and statistics.

Kolmogorov theory assumes that all small scale turbulent motions are directly related to parent large scale turbulent motions, and are statistically homogenous and isotropic (Roggemann & Welsh, 1996) This idea comes from the assumption that a turbulent atmosphere is made of various eddy sizes, but the larger eddies continuously break down into smaller eddies, uniformly distributing the energy until the viscosity of the fluid allows for complete dissipation of all kinetic energy to internal energy. Kolmogorov is then able to mathematically describe the spatial frequency characteristic of the index of refraction over a propagation path (Corley, 2010). The scale of those eddy sizes which Kolmogorov covers is referred to as the inertial subrange. Any eddy larger

then L_0 is assumed not to be homogenous in the atmosphere and any eddy smaller than l_0 dissipates the kinetic energy as heat instead of transferring it to smaller eddy. Using the inertial subrange where $\frac{1}{L_0} \ll k \ll \frac{1}{l_0}$ the power spectral density (PSD) of the index of refraction in air is

$$\Phi_n(k, z) = 0.033 C_n^2(z) k^{-\frac{11}{3}},$$

where the wavenumber, $k = \left(\frac{2\pi}{\lambda}\right)$, is related to the isotropic scale, l , by $l = \frac{2\pi}{k}$, z is the distance from the aperture, and C_n^2 is a measure of the strength of the turbulence (Andrews, 2004). Weak turbulence is associated with a C_n^2 of $10^{-17} m^{-\frac{2}{3}}$ or less, while strong turbulence generally has a C_n^2 value of $10^{-13} m^{-\frac{2}{3}}$ or greater. One issue though is that C_n^2 values vary with height above ground, so it is necessary to have altitude dependent model or profile. These profiles do exist, and have been created from extensive data collection and is based on geographic location, temperature, particle count, and many other factors. One concern however is that these profiles have primarily focused on vertical paths due to previous mission requirements, so new data is being collected, with a focus on the maritime environment and lower slant angles (Corley, 2010).

Another common parameter used to describe atmospheric turbulence is Fried's parameter, or atmospheric coherence length, r_0 . Fried's parameter describes a seeing condition at a given distance, stating that a telescopes resolution is diffraction limited at any diameter larger than r_0 (Andrews, 2004). Fried's parameter is measured in centimeters, and is formulated as

$$r_0 = \left[0.42 \sec(\zeta) k^2 \int_0^L C_n^2(z) dz \right]^{-\left(\frac{3}{5}\right)}$$

where ζ is the zenith angle, L is the distance from the source to telescope aperture along the primary axis, and k is the wavenumber. Values of 5 cm or less represent strong turbulence, where any values over 25 cm represent very weak turbulence, and good seeing conditions (Wilcox, 2009)

Kolmogorov, Fried, and C_n^2 adequately describe the spatial structure of the atmosphere and its turbulence, however it does not account for the time varying nature of the atmosphere experienced with long dwell times or slewing engagements. While there are many ways to simulate or characterize the time varying nature of atmospheric turbulence, the Greenwood time constant is a good measure of the temporal coherency of the atmosphere. Greenwood measures a time constant over which a propagation path can be assumed to have a constant index of refraction, before needing to be recalculated. The time constant is measured by

$$\tau_0 = \frac{\cos^{\frac{3}{5}}(\zeta)}{\left[2.91k^2 \int_{h_0}^H C_n^2(h) V^{\frac{5}{3}}(h) dh\right]^{\frac{3}{5}}} \approx \frac{0.314r_0}{V(h)} \left(\frac{D}{r_0}\right)^{\frac{1}{6}}$$

where $V(h)$ is the wind velocity as a function of altitude, but can also represent the slewing rate of an moving engagement scenario.

In order to analyze the effects atmospheric turbulence is having on a beam or wavefront it is useful to generate a mathematical representation, rather than just a discrete set of values. Since it is assumed a wavefront is generally continuous over a two dimensional surface wavefront for certain atmospheric conditions, it makes sense to express the wavefront as a linear combination of polynomial based functions which are orthonormal over an entire optical surface. The set of linear polynomials used in this research are Zernike polynomial represented by

$$Phase(\rho, \theta) = \sum_{i=1}^m a_i Z_i(\rho, \theta)$$

where m is the number of desired modes and a_i is the weight associated with each mode (Allen, 2007). The accuracy of a Zernike polynomial representation is a function of both the amount of disturbance in the wavefront and how many modes are chosen for the Zernike polynomial to represent the disturbance. Theoretically as m approaches infinity you can model any continuous wavefront, however it is beneficial to choose the correct number of modes based on two criteria: how turbulent is the data itself and how much accuracy can your sensors or controllers actually use.

B. MARITIME ENVIRONMENT AND DEEP TURBULENCE

Moving to the maritime environment creates compounded errors when compared the traditional ground-to-space operating procedures. First is the effect of a significantly longer propagation distance through the atmosphere. Secondly, models already created based on C_n^2 measurements were for vertical, versus horizontal, paths. These problems combined lead to a need for more robust adaptive optics control.

As previously discussed by Melissa Corely (Corely, 2010), there is work currently underway to create the new atmospheric disturbance profiles, but there is no model for horizontal paths that matches the familiarity and accuracy of the Kolmogorov model for vertical paths (Corely, 2010). Agencies and organizations like SPAWAR in San Diego, NPS in Monterey, the Naval Research Lab (NRL), University of Puerto Rico, University of Florida, and Michigan Tech have all begun work to develop and characterize parameters like C_n^2 and r_0 for horizontal paths over bodies of water. NPS has even developed an initial model, the Navy Surface Layer Optical Turbulence (NSLOT) model which models as a function of local air and sea temperatures.

While work is being done to create accurate models, that is only one of the many issues with maritime environment beam propagation. The presence of higher amounts of aerosols, large humidity and temperature fluctuations, and wave motion all contribute to higher levels of scintillation. Scintillation is intensity fluctuations or random changes in the index of refraction, and this causes a large problem with adaptive optics. One it degrades the quality and energy density on target, but it also makes it difficult for sensors to accurately estimate the wavefront and use it for correction.

C. BRANCH POINTS AND BRANCH CUTS

The scintillation effects and intensity dropouts, experienced in the presence of deep turbulence, create nulls in the intensity making it impossible for a wavefront sensor to obtain a reading at those points. These null readings in intensity are also shown to lead to a non-singular valued function when calculating the phase. These non-singular value functions mean when a closed-loop path around a subjective point is calculated, one gets a non-zero value, which is also known as a branch point. Branch points have either a net

negative or positive discontinuity, and negative branch points must be connected to a corresponding positive branch point, through a 2π discontinuity known as a branch cut.

Branch points and branch cuts drastically decrease the efficiency with which a traditional AO system can measure and compensate. That is where the focus of this research will be; attempting to accurately handle and model branch points and intensity dropouts in wavefront estimation for an AO system. Classical wavefront reconstructors use a least squares fit method, which in a sense ignores the hidden phase and glances over branch points and branch cuts. A least mean squares reconstruction method can be viewed as forming an estimate of a phase point through adding up the phase differences along every possible path from a reference point to the desired point and averaging them together (Fried, 1998). This method works well in a noise-weighted sense, however in the presence of branch points different paths can lead to different values, both of which can be equally correct. The issue is that branch points in least squares estimates of the wavefront have a degrading effect over a wide area of the aperture surrounding the branch points, as multiple paths will go through this region. Figure 1 shows the effects of branch points and branch cuts on a path following understanding of least mean squares estimators.

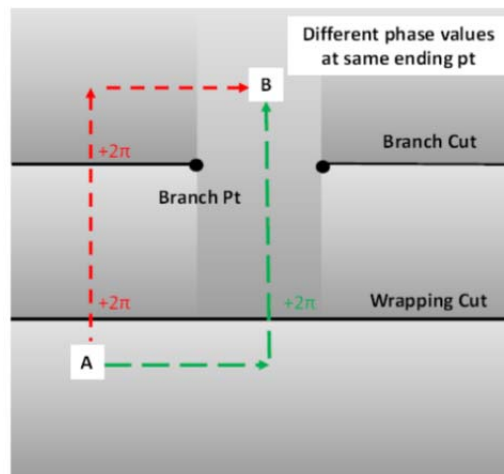


Figure 1. Phase calculation paths leading to different phase values (From Pellizzari 2010)

While it may seem incorrect at first to arrive at two different phase values for the same point, they are both equally correct. The error stems from an assumption made by least squares estimators, and that is that the measured phase differences are purely a function of a gradient of phase function. The measured phase differences need to be treated similarly as an electromagnetic field with flowing electricity, and that is as a sum of a scalar potential plus the curl of a vector potential (Fried, 1998). This view of the phase gradient as containing both a real and imaginary part goes back to the governing equations for electromagnetic propagation known as Maxwell's equations. A new reconstruction method is explored, which takes into account the hidden phase aspect, and identifies the existence and location of branch points and branch cuts in its wavefront calculations. When branch points occur, it is ideal for them to be located in areas of low intensity of the aperture to minimize the effects.

III. ADAPTIVE OPTICS

Adaptive optics technology is a multidisciplinary field which combines expertise from engineers and scientist in wide range of fields including mechanics, optics, materials science, chemistry, physics, and control theory (Allen, 2007). The primary theories associated with adaptive optics have not changed over the years; however they are becoming more refined and focused. Initially the components used in adaptive optics like cameras, deformable mirrors, and computing capabilities were very coarse by today's standards and only a certain level of control precision could be detected and applied. With large advances in this same technologies allowing for greater precision and detail, compounded with the introduction of new technologies like segmented mirrors, the requirements for advanced adaptive optics systems is growing rigorously. New developments in technology and theory are leading scientists and engineers to find new applications for AO, which in turn creates even more specific demands for an AO system; revolving in a continuous loop.

The goal of an AO system is to actively improve the capability of an optical system in the presence of imperfect operating conditions. An AO system continuously attempts to actively compensate for two main sources of optical aberrations: imperfections in the surface of a lens in the optical system and wave front distortion produced by Earth's atmosphere. AO was originally used for space imaging systems to handle the aberrations induced by Earth's atmosphere. Now AO is being employed in DE energy applications for both weapons and communications, in which coherency of the wavefront phase is necessary for precision and energy density.

A. TYPICAL ADAPTIVE OPTICS SYSTEMS

Most adaptive optics system follows a setup based on the schematic show in Figure 2, which consist of three components. The first is a wavefront sensor, which provides some measurement of the optical wavefront; that can either be discrete slope values or phase values at given points. The second component is the computer controller which attempts to reconstruct, or estimate, the phase of the wavefront and compute a

control signal for the deformable mirror. The third component is a deformable mirror, which a membrane like optical surface mounted on tiny electrical-actuators which can be used to deform the mirror to change the phase of the reflecting wavefront. The wavefront sensor and computer controller work together to, as accurately as possible, estimate what the wavefront is and figure out which points need to be either delayed or advanced so that the wavefront is as flat as possible. This research will be focusing on the sensor and computer controller portion of the AO system and the specifics of the sensor will be introduced in the following chapter.

It is important to note that the reference beam can come from a wide array of sources; however for astronomical applications it is common to use a known star. A star imaged from an extremely far distance allows it to appear as a point source, so it is easy to determine if any blurring or aberrations are present, and allow for correction.

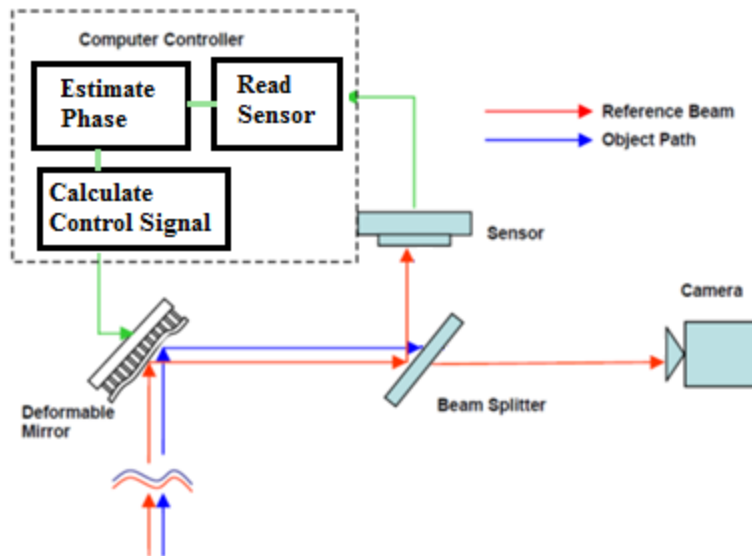


Figure 2. Typical AO System Setup (From Allen 2007)

B. BASIC ADAPTIVE OPTICS CONTROL CONCEPTS

While this research focuses on the wavefront reconstruct portion of AO system, in order to have a full understanding of the motivation behind the problem, it requires a

basic understand of the control principals associated with it. While the control computer works with the wavefront sensor to determine the errors, in order for the computer to compute a control signal it needs some sort of relationship between the deformable mirror and the wavefront. If one considers a classical negative feedback system illustrated in Figure 3 then the wavefront sensor can be interpreted as a state estimator, the optical wavefront is the “plant” that is controlled, and the deformable mirror is the actual control element (Tyson, 2004). In Figure 2 there is also a reconstructor block, and this is what provides that relationship between the deformable mirror and the wavefront sensor.

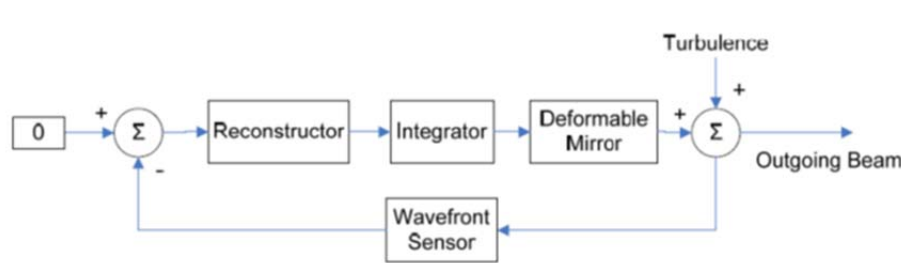


Figure 3. Classical AO Control System

In a simple overview a wavefront estimator, which can a multitude of different sensors, represents the wavefront as a vector of discrete values. These values can either be local slope measurements, local phases, or even coefficients associated with equations like Zernike polynomials. Recalling that a deformable mirror has a discrete number of actuators, either electrostrictive (PZT) or magnetostrictive (PMN), a few other assumptions need to be made as well. In modern AO systems it is assumed there is static coupling between actuators, however there is not dynamic coupling in the deformable mirror (DM). The response time of a DM is so quick the relationship between voltage and actuator movement is treated as a step response. It is also important to note that the relationship between voltage and actuator movement can be considered linear with a small displacement range. With these assumptions the concept of a poke, or influence, matrix can be constructed.

The deformable mirror has a discrete amount of actuators and the wavefront sensor outputs a discrete amount of coefficients or slopes/phases, so each actuator is fully

triggered and the corresponding change to the outputs of the wavefront sensor are logged. This is done sequentially and individually for each actuator, while holding all others at zero. These measurements are then used to create a poke matrix whose columns correspond to wavefront sensor changes associated with that actuator, and the rows correspond to the same coefficient or slope reading across each actuator. The structure of a typical poke matrix using a Shack-Hartmann wavefront sensor (SHWFS), which outputs local x and y sloped measurements at discrete points, is:

$$\Gamma = \begin{bmatrix} x_{11} & \dots & x_{1k} \\ x_{21} & \dots & x_{2k} \\ \vdots & \dots & \vdots \\ x_{n1} & \dots & x_{nk} \\ y_{11} & \dots & y_{1k} \\ y_{21} & \dots & y_{2k} \\ \vdots & \dots & \vdots \\ y_{n1} & \dots & y_{nk} \end{bmatrix}$$

where n is the number of discrete x and y slope measurements and k is the number of actuators on the deformable mirror. In the case of a sensor with n slope measurement points, and k actuators the matrix is 2n x k. Using the influence matrix the relationship between command voltage, \mathbf{c} , and slope output, \mathbf{s} , from the sensor is defined as

$$\mathbf{s} = \Gamma \mathbf{c}$$

Given the control architecture of AO system, usually the slope error is known, and a actuator command voltage is desired. Normally the multiple-input-multiple-output (MIMO) AO system has more inputs (sensor readings) than outputs (actuator commands), and is therefore over determined and creating a non-square matrix. Taking the pseudo inverse of the poke matrix, Γ^+ , and pre-multiplying the slope error measurements creates a least squares solution to the minimization of the wavefront error correction:

$$\Gamma^+ \mathbf{s} = \Gamma^+ \Gamma \mathbf{c} = \mathbf{c}$$

In Figure 3 the controller implemented is a simple integrator, which uses a gain, g , to iteratively calculate a new control using the following law

$$\mathbf{c}_{new} = \mathbf{c}_{old} - g\Gamma^+ \mathbf{s}$$

This is just one example of a simple controller, but any control method can be implemented.

IV. WAVEFRONT ESTIMATION

The ability to reconstruct and estimate the wavefront is essential for a successful AO system. In traditional wavefront sensing for optical purposes, the phase determination can take anywhere from hours to days, however adaptive AO systems operate at a higher demand. AO systems are used for real-time corrections and need to operate at rates up to several hundred of hertz and spatial resolution as fine as 1/100 of the aperture diameter. There are two methods for sensing a wavefront; direct and indirect approach (Tyson, 1998). The direct approach takes sensor data and attempts to explicitly solve for the phase of the wavefront which is used as information for correcting unwanted components. The indirect method directly relates the sensor data to a control signal, bypassing the need to explicitly solve for the phase (Tyson, 1998). While the indirect may be quicker, the direct method provides greater detail of the errors in the wavefront, therefor determination of the method used is completely situation dependent.

A. WAVEFRONT SENSOR

There are various sensors used for wavefront detection; this research makes use of a Shack-Hartmann wavefront sensor (SHWFS) design. A SHWFS consist of two dimensional array of lenslet placed in a grid pattern in front of an imaging sensor, as shown in Figure 3.

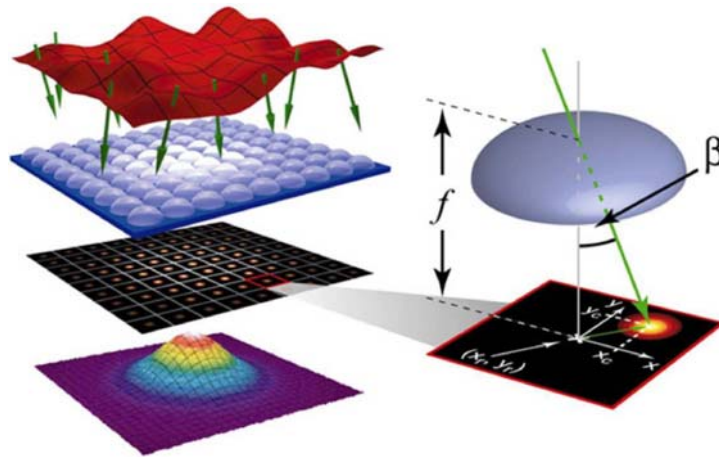


Figure 4. SHWFS Schematic (From Mann 2012)

Each lenslet acts as an aperture, causing the incoming light to converge into spots on the CMOS sensor. The location of each spot is directly proportional to the average slope of the wavefront passing through the corresponding lenslet (Primot, 2003). A perfectly coherent wavefront is used to produce a reference image for centroid locations. The deviation of each centroid for an aberrated wavefront from each reference location in the x and y directions is directly proportional to the local x and y slopes, α and β , respectively. Figure 4 shows the relationship between the slope of the wavefront and the displacement of the centroid. The displacement of the centroid is related to the slope by:

$$\alpha_{i,j} = \frac{2\pi}{\lambda f} \Delta x_{i,j}$$

$$\beta_{i,j} = \frac{2\pi}{\lambda f} \Delta y_{i,j}$$

where λ is the wavelength of light passing through, f is the focal length of the lenslet array, and Δx and Δy are the shifts of the centroids in each dimension, respectively (Zhu, Sun, Bartsch, Freeman & Fainman, 1999).

The centroids are calculated for each array of pixels corresponding to the equivalent sub-lenslet. The Centroid calculation is done through a summation of the pixel location from the given axis weighted by the corresponding pixel intensity, defined as

$$x_{cent}(i,j) = \frac{\sum_{u=1}^m \sum_{v=1}^m u * I(u,v)}{\sum_{u=1}^m \sum_{v=1}^m I(u,v)}$$

$$y_{cent}(i,j) = \frac{\sum_{u=1}^m \sum_{v=1}^m v * I(u,v)}{\sum_{u=1}^m \sum_{v=1}^m I(u,v)}$$

where i and j specify the given lenslet and u and v correspond to the local pixel indexes of the given subaperture with (1,1) being the bottom left corner.

The algorithm is carried out on a small section of pixels from the CMOS sensor, corresponding to the given lenslet. It is assumed the very small gaps between the circular lenslet arrays create minimal noise on the CMOS sensor through leakage, therefore it can be ignored. With the corresponding slope measurements then either a control signal can be calculated for the DM, using the indirect control method, or the phases of the wavefront can be reconstructed if a direct approach is used.

B. SHWFS OUTPUT SIMULATION METHOD

The research was conducted in an entirely simulated environment, so a method for simulating the SHWFS is needed. Using WAVEPROP, developed by the Optical Sciences Company (tOSC), to simulate the effects of a turbulent environment on a beam, through the use of Kolmogorov statistics, an incident phase and amplitude screen is created. These phase and amplitude screens simulate what will be seen at the face of a SHWFS. The SHWFS is a planar array of multiple lenses with a sensor placed at the focal point; electro-optical theory can be used to simulate the effects of each lenslet. In wave optics each lens can be modeled with a mathematical expression of a phase device (Schmidt, 2010)

$$\Phi_{lens}(x, y) = w(x, y) \exp \left[-i \left(\frac{k}{2f} \right) (x^2 + y^2) \right]$$

where k is the wave number, f is the focal length and $w(x, y)$ represents the aperture function

$$w(x, y) = f(x) = \begin{cases} 1, & |x| < \frac{d}{2}, |y| < \frac{d}{2} \\ 0, & other \end{cases}$$

where d is the size of the lens. Fourier optics provides a simple calculation to evaluate the intensity distribution at the focal plane as

$$\begin{aligned} I(x_f, y_f) &\propto \left| \iint_{-\infty}^{\infty} \psi_1(x, y) w(x, y) \exp \left[-i \left(\frac{k}{f} \right) (xx_f + yy_f) \right] dx dy \right|^2 \\ &= |FT\{\psi_1(x, y) w(x, y)\}|^2 \end{aligned}$$

where ψ is the complex wave distribution of the phase and amplitude screen at the face of the lenslet array defined as

$$\psi(x, y) = A(x, y) \exp(i * \theta(x, y))$$

With $A(x, y)$ being the amplitude and $\theta(x, y)$ being the phase screens, respectively. While Fourier optics provide a method of simulating the optical effects of each lens, it has been proven that the equivalent slope output through a centroiding algorithm of a SHWFS is directly proportional to the intensity weighted average of the gradient between each pixel across the entire sensor (Fried 1997). This relates to the centroiding algorithm used to calculate the displacement of a SHWFS, in which areas of higher intensity have a greater

effect on the centroid location. The equivalent x and y gradient, or tilt, can be calculated for each subaperture as

$$S_x(i, j) = \sum_{u=1}^{m-1} \sum_{v=1}^m \frac{\left(\arg(\psi(u+1, v)\psi^*(u, v)) \right) \left(\text{abs}(\psi(u+1, v)\psi^*(u, v)) \right)^2}{\sum_{u=1}^{m-1} \sum_{v=1}^m \left(\text{abs}(\psi(u+1, v)\psi^*(u, v)) \right)^2}$$

$$S_y(i, j) = \sum_{u=1}^m \sum_{v=1}^{m-1} \frac{\left(\arg(\psi(u, v+1)\psi^*(u, v)) \right) \left(\text{abs}(\psi(u, v+1)\psi^*(u, v)) \right)^2}{\sum_{u=1}^m \sum_{v=1}^{m-1} \left(\text{abs}(\psi(u, v+1)\psi^*(u, v)) \right)^2}$$

with u and v corresponding to the indices of the phase and amplitude screens which correspond to the area covered by the given lenslet and

$$\arg(x) = \text{atan}\left(\frac{\text{Im}(x)}{\text{Re}(x)}\right)$$

A simple test algorithm was run to ensure the proper functioning of the SHWFS simulator. A phase plane shown in Figure 5 is defined by the equation

$$\theta(x, y) = x^2 + y^3$$

which generated the following phase screen

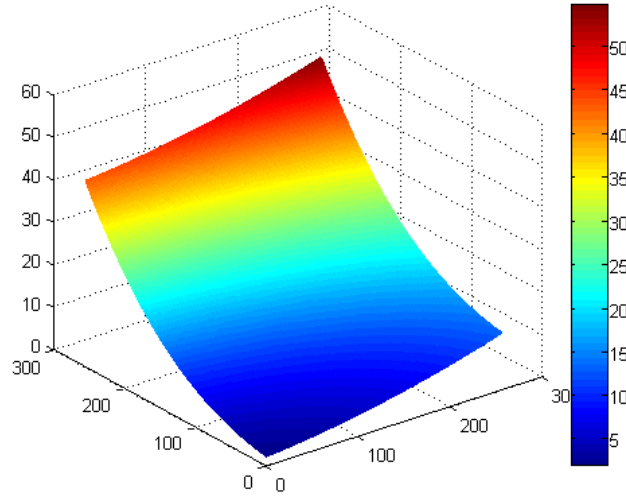


Figure 5. SHWFS Slope Simulate Test Phase Screen ($z=x^2+y^3$)

The weighted averaging was used to simulate the slopes generated by a 64x64 lenslet SHWFS, and Figure 6 shows the results.

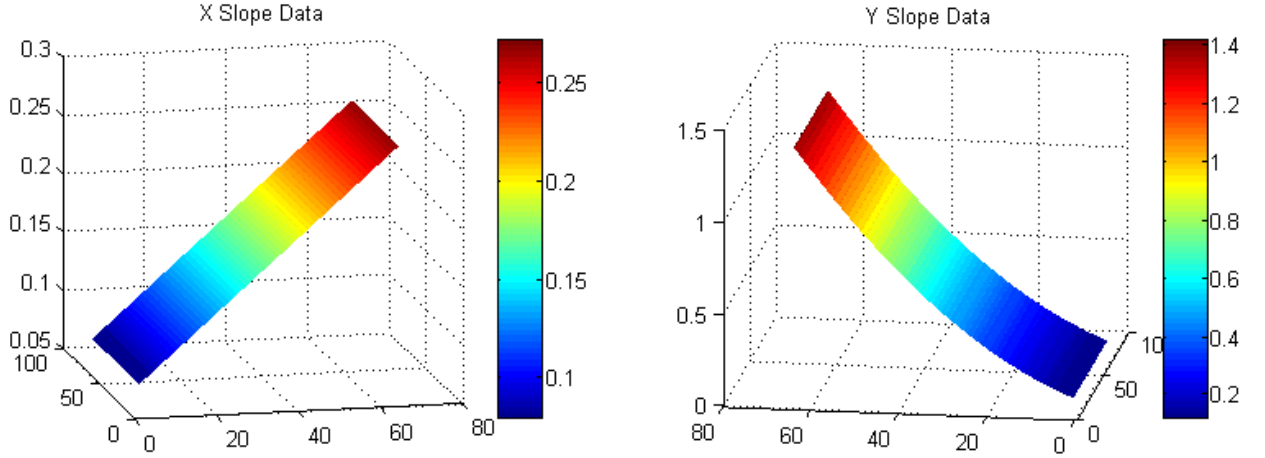


Figure 6. SHWFS Output Simulation for X and Y slopes based on Fig. 5 Phase

From basic mathematical differentiation of the phase equation, it makes sense that the x slope follows a linear path and y data follows a quadratic (second order) growth path. A normally distributed, random amplitude screen was generated and applied for the algorithm.

C. ESTIMATION (RECONSTRUCTION) METHODS

As discussed in section III there are two methods of actually implementing the slope data from a SHWFS for control of a DM, either direct or indirect, and wavefront estimation is used with the direct method. There are numerous methods to actually go from slope data to an estimated phase. Two common methods, typically used in low turbulence environment, are zonal and modal estimation (Baker, 2005). Zonal attempts to estimate discrete phase points in each zone, while modal attempts to solve for the coefficients of polynomial expansion like functions. A third method, the focus of this research, is a noise-variance weighted complex exponential reconstructor, which is a recursive noise-weighted averaging algorithm which use complex phase differentials instead of just slopes.

First the different geometries need to be introduced which relate the slope measurements of the SHFWS to the calculated phase points. Figure 7 shows the differing geometries where the lines indicated the slope or phase difference values and the dots represent the calculated phase values. Zonal and Modal reconstruction methods operate

on Southwell geometry, while the NVWCER operates on a combined Hudgin and Fried geometry.

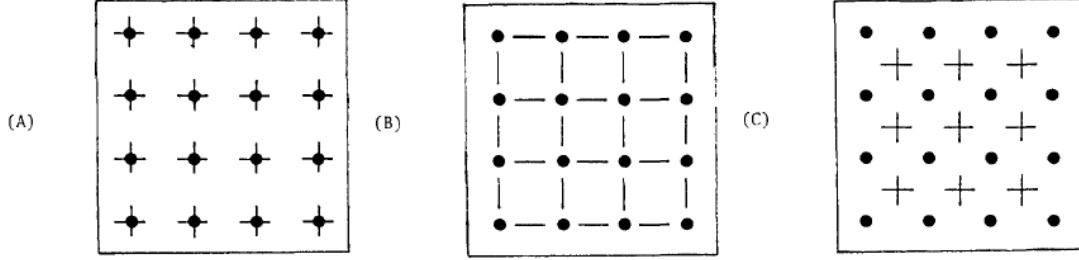


Figure 7. Common Reconstruction Geometries (a) Southwell (b) Hudgin (c) Fried (From Southwell 1980)

1. Zonal Reconstruction

Using a SHWFS produces incident x and y slopes at a single aperture, so the geometric configuration in Figure 7(a) is used. The first step is to formulate a relationship between the phases and slopes. If quadratic interpolation is used to define the dependence of the phase in the x and y direction then the following two polynomials are created.

$$\begin{aligned}\phi &= c_0 + c_1x + c_2x^2 \\ \phi &= c_0 + c_1y + c_2y^2\end{aligned}$$

with relationship for the dependency of the phase on the x and y locations, the differentiation of each equation should provide the equivalent dependency equations for the x and y slopes. Taking the derivative of the equations with respect to x and y yields

$$\begin{aligned}S_x &= c_1 + 2c_2x \\ S_y &= c_1 + 2c_2y\end{aligned}$$

Since two slope measurements are given for each phase point it allows for the determination of c_1 and c_2 . A relationship between adjacent phases and slope measurements is created; the average slope of two adjacent phase points is equal to the difference of the two phase points divided by the length h (Southwell, 1980).

$$\frac{S_{i+1,j}^x + S_{i,j}^x}{2} = \frac{\phi_{i+1,j} - \phi_{i,j}}{h}, i = 1:(N-1); j = 1:N$$

$$\frac{S_{i,j+1}^y + S_{i,j}^y}{2} = \frac{\phi_{i,j+1} - \phi_{i,j}}{h}, i = 1:N ; j = 1:(N-1)$$

Each lenslet represents an equal sub-area of the entire aperture, h^2 , so $h=D/N$ where D is the diameter of the entire aperture and N is the number of lenslets along one side. In order to simultaneously solve all for all phase points in a least squares solution, a matrix interpretation of the operation is created. First a vector of the slopes, of length $2N^2$, is created with the x slopes first, then y slopes.

$$\bar{S} = \begin{bmatrix} S_1^x \\ S_2^x \\ \vdots \\ S_{N^2}^x \\ S_1^y \\ S_2^y \\ \vdots \\ S_{N^2}^y \end{bmatrix}$$

Next a sparsely populated matrix D, of size $2N(N-1)$ by $2N^2$, is created which performs the adjacent slope averaging. Then a matrix A, of size $2N(N-1)$ by N^2 , is created which calculates the adjacent phase differences. Using matrix formulation the phase and slopes are related by

$$[D]\bar{S} = [A]\bar{\phi}$$

Given that D, \bar{S} , and A are known the phases can be calculated by multiplying both sides by the inverse of A. However, since there are more slope measurements than phase points the equation is over determined and A is not a square matrix, so the pseudo inverse must be taken, which results in a least squares solution

$$\bar{\phi} = [A]^\dagger [D]\bar{S}$$

where

$$[A]^\dagger \stackrel{\text{def}}{=} [A]^T ([A][A]^T)^{-1}$$

2. Modal Reconstruction

In modal reconstruction the wavefront is described with coefficients of multiple different optical modes, expressed as an optical wavefront expansion; similar to that of a polynomial expansion. It makes use of Southwell geometry, like modal reconstruction, assuming coincident slope measurements. There are numerous mathematical models to

represent the different modes, and this thesis focuses on the use of Zernike polynomials. Zernike polynomials have two key characteristics: they are orthogonal over a unit disk and also expandable to a limitless degree. The orthonormality of the polynomials means that their derivative also exists over the entire unit disc, which becomes useful when attempting to fit slope data. Zernike polynomials represent a wavefront a desired m and n degree through (Tyson and Frazier, 2004)

$$\phi(r, \theta) = A_{00} + \frac{1}{\sqrt{2}} \sum_{n=2}^{\infty} A_{n0} \mathfrak{R}_n^0\left(\frac{r}{R}\right) + \sum_{n=1}^{\infty} \sum_{m=1}^n (A_{nm} \cos(m\theta) + B_{nm} \sin(m\theta)) \mathfrak{R}_n^m\left(\frac{r}{R}\right)$$

where

$$\mathfrak{R}_n^0\left(\frac{r}{R}\right) = \sum_{s=0}^{\frac{n-m}{2}} \frac{(-1)^s (n-s)!}{s! \left(\frac{n+m}{2} - s\right)! \left(\frac{n-m}{2} - s\right)!} \left(\frac{r}{R}\right)^{n-2s}$$

It is important to note that Zernike polynomials exist in polar coordinates and exist over a unit disk. The radius is normalized with the $\left(\frac{r}{R}\right)$ term, where r is the radius at a given point and R is the radius of the entire aperture. The first 21 modes are listed in Table 1.

#	n	m	Polynomial
0	0	0	1
1	1	1	$\rho \cos[\theta]$
2	1	1	$\rho \sin[\theta]$
3	1	0	$-1 + 2 \rho^2$
4	2	2	$\rho^2 \cos[2 \theta]$
5	2	2	$\rho^2 \sin[2 \theta]$
6	2	1	$\rho (-2 + 3 \rho^2) \cos[\theta]$
7	2	1	$\rho (-2 + 3 \rho^2) \sin[\theta]$
8	2	0	$1 - 6 \rho^2 + 6 \rho^4$
9	3	3	$\rho^3 \cos[3 \theta]$
10	3	3	$\rho^3 \sin[3 \theta]$
11	3	2	$\rho^2 (-3 + 4 \rho^2) \cos[2 \theta]$
12	3	2	$\rho^2 (-3 + 4 \rho^2) \sin[2 \theta]$
13	3	1	$\rho (3 - 12 \rho^2 + 10 \rho^4) \cos[\theta]$
14	3	1	$\rho (3 - 12 \rho^2 + 10 \rho^4) \sin[\theta]$
15	3	0	$-1 + 12 \rho^2 - 30 \rho^4 + 20 \rho^6$
16	4	4	$\rho^4 \cos[4 \theta]$
17	4	4	$\rho^4 \sin[4 \theta]$
18	4	3	$\rho^3 (-4 + 5 \rho^2) \cos[3 \theta]$
19	4	3	$\rho^3 (-4 + 5 \rho^2) \sin[3 \theta]$
20	4	2	$\rho^2 (6 - 20 \rho^2 + 15 \rho^4) \cos[2 \theta]$

Table 1. First 21 Zernike Polynomial Modes (From Wyant 2003)

The SHWFS modeled in this research relies on a square grid arrangement which introduces two issues; the need for a proper coordinate system and fitting a circle to the square. Zernike polynomials can be transformed into Cartesian coordinates through the use of two relationships,

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \arctan\left(\frac{y}{x}\right)$$

and their equivalent equations are show in Table 2. Similarly to polar coordinates, Cartesian coordinates are normalized as well where x and y are both normalized to satisfy

$$\left[\rho = \sqrt{x^2 + y^2}\right] < 1$$

#	n	m	Polynomial	Term
0	0	0	1	Piston
1	1	1	x	X-Tilt
2	1	1	y	Y-Tilt
3	1	0	$-1 + 2(x^2 + y^2)$	Focus
4	2	2	$x^2 - y^2$	Astigmatism plus defocus
5	2	2	$2xy$	Astigmatism plus defocus
6	2	1	$-2x + 3x(x^2 + y^2)$	Coma
7	2	1	$-2y + 3y(x^2 + y^2)$	Tilt
8	2	0	$1 - 6(x^2 + y^2) + 6(x^2 + y^2)^2$	Third-Order Spherical and Focus
9	3	3	$x^3 - 3xy^2$	Fifth-Order Aberration
10	3	3	$3x^2y - y^3$	Fifth-Order Aberration
11	3	2	$-3x^2 + 3y^2 + 4x^2(x^2 + y^2) - 4y^2(x^2 + y^2)$	Fifth-Order Aberration
12	3	2	$-6xy + 8xy(x^2 + y^2)$	Fifth-Order Aberration
13	3	1	$3x - 12x(x^2 + y^2) + 10x(x^2 + y^2)^2$	Fifth-Order Aberration
14	3	1	$3y - 12y(x^2 + y^2) + 10y(x^2 + y^2)^2$	Fifth-Order Aberration
15	3	0	$-1 + 12(x^2 + y^2) - 30(x^2 + y^2)^2 + 20(x^2 + y^2)^3$	Fifth-Order Aberration
16	4	4	$x^4 - 6x^2y^2 + y^4$	Seventh-Order Aberration
17	4	4	$4x^3y - 4xy^3$	Seventh-Order Aberration
18	4	3	$-4x^3 + 12xy^2 + 5x^3(x^2 + y^2) - 15xy^2(x^2 + y^2)$	Seventh-Order Aberration
19	4	3	$-12x^2y + 4y^3 + 15x^2y(x^2 + y^2) - 5y^3(x^2 + y^2)$	Seventh-Order Aberration

Table 2. First 21 Zernike Polynomials in Cartesian Coordinates (From Allen 2007)

In order to properly fit the square sensor array inside of a unit circle the SHWFS had to be circumscribed inside of the unit circle. Assuming an $N \times N$ square sensor grid, the circle was created with a radius $R = \sqrt{2N^2}$ as shown in Figure 8.

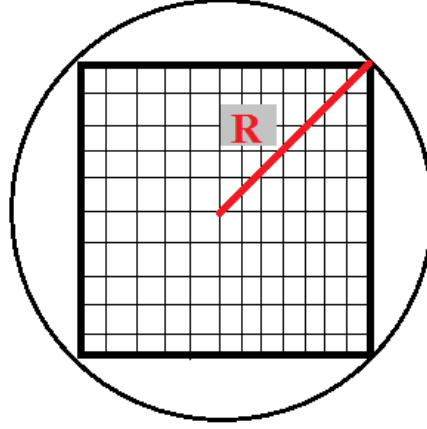


Figure 8. SHWFS Grid Circumscribed inside Unit Disk for Zernike Polynomial Evaluation

Recalling back from section II.A, and the recent explanation of Zernike polynomials it is now possible to define any wavefront as a summation of m modes, multiplied by its corresponding coefficient a_j

$$\phi(x, y) = \sum_{j=1}^m a_j Z_j(x, y)$$

where Z_j represents the j th Zernike mode. Similarly to the zonal reconstruction methods, it is possible to formulate a matrix version of Zernike mode summation

$$\bar{\phi} = [Z]\bar{a}$$

where $\bar{\phi}$ is N^2 length vector, \bar{a} is a m length vector corresponding to the mode coefficients, and $[Z]$ is a N^2 by m size matrix corresponding the evaluation of each Zernike mode at each phase location. Using the pseudo inverse of $[Z]$ the coefficients, \bar{a} , can be solved in a least squares solution.

$$\bar{a} = [Z]^\dagger \bar{\phi}$$

Recalling that Zernike polynomials are orthonormal over the entire unit disk, meaning there derivatives exist over the entire unit disk; it is possible to fit the SHWFS slope data to the derivatives of the Zernike polynomials (Southwell, 1980).

$$s^x = \sum_{j=1}^m a_j \frac{\partial Z_j(x, y)}{\partial x}$$

$$S^y = \sum_{j=1}^m a_j \frac{\partial Z_j(x, y)}{\partial y}$$

The same matrix formulation can apply relating the SHWFS slope data to the Zernike coefficients through a least squares solution as

$$\bar{a} = [\partial Z]^\dagger \bar{S}$$

where ∂Z is a matrix, $2N^2$ by m , representing the derivatives of the Zernike polynomials for both x and y , and S is a vector of length $2N^2$, representing the x and y slope measurements of the SHWFS. Once the coefficients are calculated they are used to directly evaluate the phase at any point in the aperture, or corresponding to a given lenslet. The higher order modes are related to higher frequency aberrations in a wavefront, so there is a tradeoff between efficiency and accuracy, when deciding how many modes to use. Modal reconstruction does also apply a form of smoothing the phase functions, as it is a continuous function, which can introduce some errors in the phase estimation.

3. Noise-Variance Weighted Complex Exponential Reconstructor

It was discussed previously that nulls in intensity readings on the SHWFS lead to the presence of branch points, which is addressed in depth by Dr. Fried in “Branch Point problem in adaptive optics.” Branch points arise as an issue with traditional least squares reconstructor, like zonal or modal methods, because they separate the phase and amplitude when expressing the optical field. If the perturbations are expressed as a complex phasor, $u = A \exp(i\phi)$, composed of the phase and amplitude, ϕ and A , then branch point location and information are not lost during the reconstruction (Fried, 2001). The basic concept behind the NVWCER lies in the initial formulation of how to express the phases and phase differences. In the reconstruction process phases are represented as phasors, $u = \exp(i\phi)$, and phase differences are represented as differential phasors, $\Delta u = \exp(i\Delta\phi)$. The NVWCER is a recursive, multi-path averaging algorithm which uses multiple paths, from an initial phasor to another phasor, to best estimate the differences. It operates on a Hudgin based geometry grid of square size 2^N+1 . Like traditional methods multiple paths are added together and use to obtain a best estimate for

overall difference, however with phasors one multiplies to perform the equivalent of adding and multiplies by the complex conjugate for subtraction.

$$\begin{aligned}\phi_1 - \phi_2 &\sim \exp(i\phi_1) \exp(-i\phi_2) \\ \phi_1 + \phi_2 &\sim \exp(i\phi_1) \exp(i\phi_2)\end{aligned}$$

The noise-variance weighting aspect of the NVWCER utilizes the relative signal-to-noise-ratio (SNR) for each sensor measurement to weight the corresponding path versus other paths for the calculation. A quick review of statistics leads to two statements that provide the basis for noise variance weighting when combining several noise laden quantities (Fried, 2001). When noise laden quantities are added the best estimate for the value of the sum is simply the sum of the quantities, and the noise induced variance of the sum is equal to sum of the noise induced variances of each of the quantity (Fried, 2001). Secondly, the best average of several noise laden quantities is the sum of the ratios of the quantity divided by its noise induced variance – that sum is in turn divided by the sum of the inverse of the individual noise induced variances (Fried, 2001). Using the notation of for noise laden quantities and noise induced variance, x_n and σ_n^2 , respectively:

$$\begin{aligned}X_{sum} &= \sum_{n=1}^N x_n, \sigma_{sum}^2 = \sum_{n=1}^N \sigma_n^2 \\ X_{AVG} &= \frac{\sum_{n=1}^N \frac{x_n}{\sigma_n^2}}{\sum_{n=1}^N \frac{1}{\sigma_n^2}}, \sigma_{AVG}^2 = \frac{1}{\sum_{n=1}^N \frac{1}{\sigma_n^2}}\end{aligned}$$

The NVWCER is a two part process, first the differential phasors are used to reconstruct the wavefront then the branch points are analyzed and placed. The reconstruction is a three step process; reduce, solve, and rebuild stages. The reduce stage recursively goes from a $2^N + 1$ square grid to a $2^{N-1} + 1$ square grid, as it uses several differential phasors and calculates a single noise-variance weighted differential phasor corresponding to the de-resolution, until a 2x2 grid is reached. The solve stage uses the four differential phasors from the last step of the reduce stage, along with an assigned zero phase point, to calculate the phasor at each corner. The rebuild stage uses the

calculated phasors and corresponding differential phasors to calculate the new phasors working from a $2^N + 1$ to a $2^{N+1} + 1$ square grid each iteration, until the original resolution is reached. Each stage is covered in depth in the subsequent sections

a. Reduce Stage

The reduce stage of the algorithm the original 2^N+1 square grid is successively reduced to $2^{N-1}+1$, and so forth, until a 2 by 2 size grid is obtained. In each step there is also a corresponding reduction in the number of differential phasors. In a process explained below, the differential phasors, and accompanying noise induced variances, corresponding to the $2^{N-1}+1$ square grid are formed from a noise-variance weighted estimate of the 2^N+1 grid differential phasors and variances (Fried, 2001). One iteration of the reduce stage is shown in Figure 9 where grid (a) is of size $N=3$, and is reduced to grid (b) of size $N=2$.

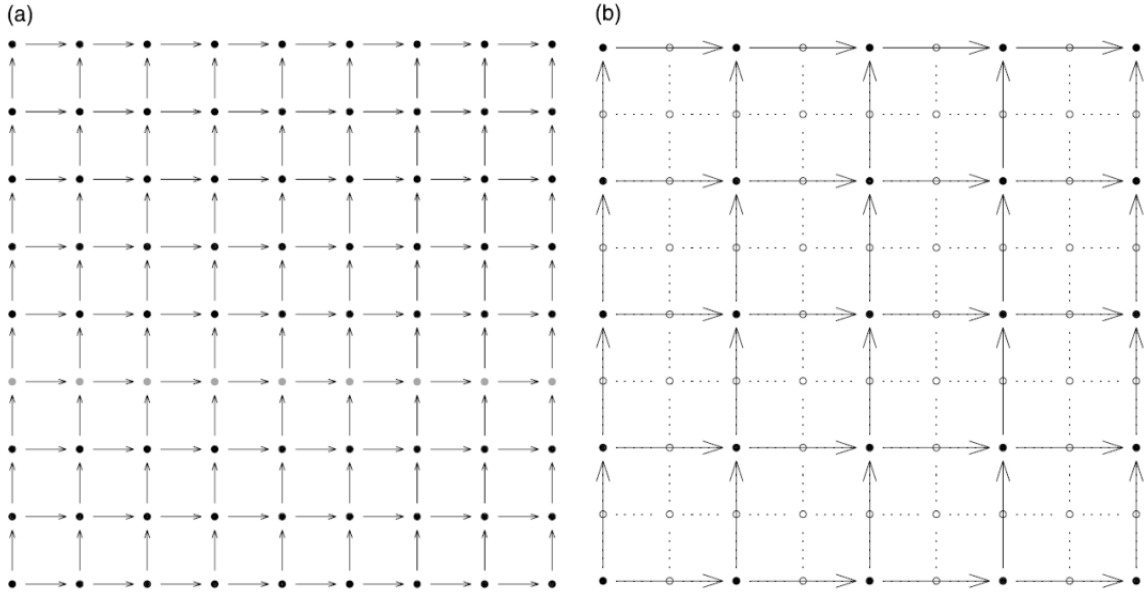


Figure 9. Lattice Points for Reduce Stage of NVWCER (a) $N=3$ (b) $N=2$
(From Fried 2001)

For each iteration in the reduction process it is necessary to calculate a value for the differential phasor and corresponding noise induced variances between adjacent grid point on the $2^{N-1}+1$ grid. The differential phasors, for either x or y, to be

calculated are referred to as $x\Delta u$ and $y\Delta u$, and the noise induced variances are $\sigma_{x\Delta u}^2$ and $\sigma_{y\Delta u}^2$. For notation purposes in the calculations a,b,c,d,e,f,g,h, and i refer to differential phasors from the originating lattice in each step of the reduction. There are six possibilities for the location when calculating the differential phasor for the reduced lattice. The first four, shown in Figure 10, are when it is located on the edge, and the second two, shown in Figure 11, occur on the interior of the lattice. The lines indicate differential phasors from the starting lattice, solid circles represent grid points on the reduced lattice, and open circles represent grid points on starting lattice.

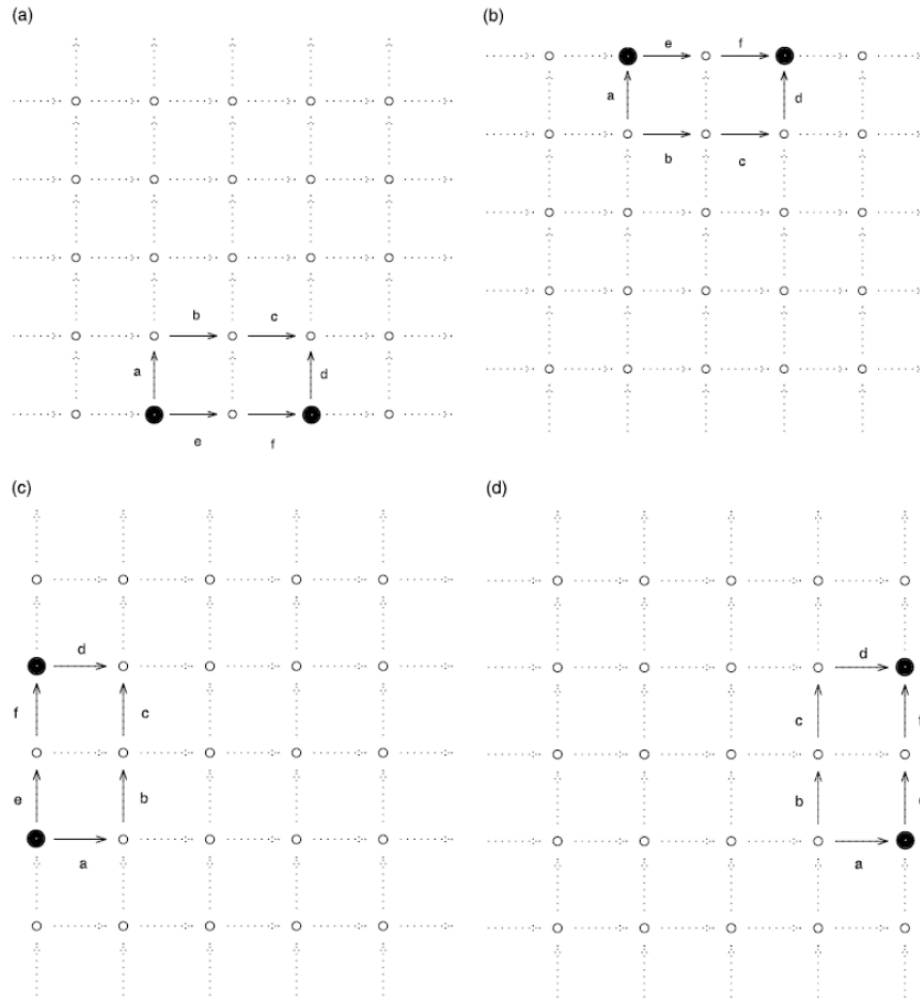


Figure 10. Calculation of Differential Phasors for Reduce stage, grid points located on edge of lattice (From Fried 2001)

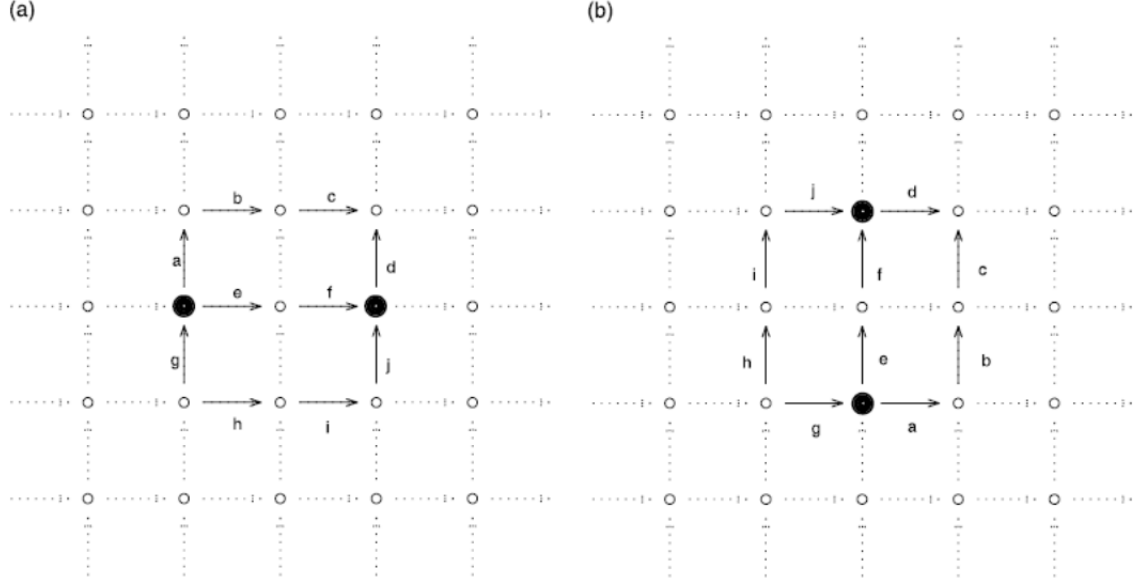


Figure 11. Calculation of Differential Phasors for Reduce stage, grid points located on interior of lattice (From Fried 2001)

When the differential phasor lies between two grid points located on the edge of the reduced lattice, the following four pairs of equations are used. It is important to note that at each step of the reduction process, the corresponding values and variances need to be store for use in the rebuild stage. For Figures 10.a

$$\sigma_{x\Delta u}^2 = \frac{1}{\frac{1}{\sigma_e^2 + \sigma_f^2} + \frac{1}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2}}$$

$$x\Delta u = \left\{ \left(\frac{ef}{\sigma_e^2 + \sigma_f^2} + \frac{abcd^*}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2} \right) \sigma_{x\Delta u}^2 \right\}$$

For Figure 10.b the following equations are used

$$\sigma_{x\Delta u}^2 = \frac{1}{\frac{1}{\sigma_e^2 + \sigma_f^2} + \frac{1}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2}}$$

$$x\Delta u = \left\{ \left(\frac{ef}{\sigma_e^2 + \sigma_f^2} + \frac{a^*bcd}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2} \right) \sigma_{x\Delta u}^2 \right\}$$

and for Figure 10.c

$$\sigma_{y\Delta u}^2 = \frac{1}{\frac{1}{\sigma_e^2 + \sigma_f^2} + \frac{1}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2}}$$

$$y\Delta u = \left\{ \left(\frac{ef}{\sigma_e^2 + \sigma_f^2} + \frac{abcd^*}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2} \right) \sigma_{y\Delta u}^2 \right\}$$

lastly for Figure 10.d

$$\sigma_{y\Delta u}^2 = \frac{1}{\frac{1}{\sigma_e^2 + \sigma_f^2} + \frac{1}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2}}$$

$$y\Delta u = \left\{ \left(\frac{ef}{\sigma_e^2 + \sigma_f^2} + \frac{a^*bcd}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2} \right) \sigma_{y\Delta u}^2 \right\}$$

When the phasors on the reduced grid lie between points on the interior of the lattice the following equations are used, based on Figure 11. For Figure 11.a

$$\sigma_{x\Delta u}^2 = \frac{1}{\frac{1}{\sigma_e^2 + \sigma_f^2} + \frac{1}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2} + \frac{1}{\sigma_g^2 + \sigma_h^2 + \sigma_i^2 + \sigma_j^2}}$$

$$x\Delta u = \left\{ \left(\frac{ef}{\sigma_e^2 + \sigma_f^2} + \frac{abcd^*}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2} + \frac{g^*hij}{\sigma_g^2 + \sigma_h^2 + \sigma_i^2 + \sigma_j^2} \right) \sigma_{x\Delta u}^2 \right\}$$

and for Figure 11.b

$$\sigma_{y\Delta u}^2 = \frac{1}{\frac{1}{\sigma_e^2 + \sigma_f^2} + \frac{1}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2} + \frac{1}{\sigma_g^2 + \sigma_h^2 + \sigma_i^2 + \sigma_j^2}}$$

$$y\Delta u = \left\{ \left(\frac{ef}{\sigma_e^2 + \sigma_f^2} + \frac{abcd^*}{\sigma_a^2 + \sigma_b^2 + \sigma_c^2 + \sigma_d^2} + \frac{g^*hij}{\sigma_g^2 + \sigma_h^2 + \sigma_i^2 + \sigma_j^2} \right) \sigma_{y\Delta u}^2 \right\}$$

This provides the necessary equations for carrying out the reduce stage of NVWCER from a 2^N+1 square grid to a 2 by 2 square grid. Connecting the equations with the paths laid out in Figure 10 and Figure 11, along with the mention of statistical averaging, it is easy to follow formulation of each equation.

b. Solve Stage

The reduce stage ends with four differential and phasors and four noise-induced variances, corresponding to the outer four edges of the lowest resolution lattice. If a phasor value of 0 is assigned to one of the corners as the reference point, then a least squares noise-variance weighted solution for the phasor at each corner can be calculated. This method however is intended for use with phases and adapted to work with phasors,

so a method is implemented to work with phasors directly and avoid any 2π ambiguities (Fried, 2013). The goal is to determine four complex value phasors, with unity magnitude, corresponding to the four corner points, of the grid show in Figure 12.

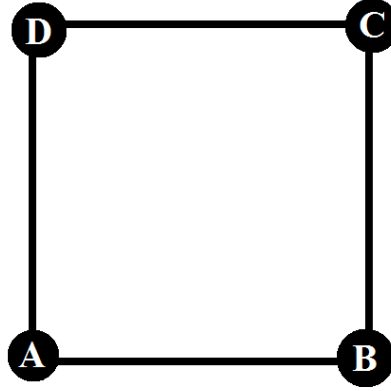


Figure 12. Grid Patter for Solve Stage of NVWCER

The four phasors, analogous to A, B, C, and D will be referred to as $\mathbb{P}_A, \mathbb{P}_B, \mathbb{P}_C$, and \mathbb{P}_D and the differential phasors from the reduce stage are $u\Delta_{BA}, u\Delta_{CB}, u\Delta_{CD}$, and $u\Delta_{DA}$. The relationship between the phasors and measured differential phasors is as follows (Fried, 2013):

$$\begin{aligned} u\Delta_{BA} &= \frac{\mathbb{P}_B}{\mathbb{P}_A} \\ u\Delta_{CB} &= \frac{\mathbb{P}_C}{\mathbb{P}_B} \\ u\Delta_{CD} &= \frac{\mathbb{P}_D}{\mathbb{P}_C} \\ u\Delta_{DA} &= \frac{\mathbb{P}_D}{\mathbb{P}_A} \end{aligned}$$

Dr. Fried makes note though, that since our differential phasors have a noise induced variance, they are only imperfectly known, so a path for example corresponding from A to D yields

$$u\Delta_{DA} \approx u\Delta_{BA}u\Delta_{CB}(u\Delta_{CD})^*$$

In order to accurately account for these imperfect measurements, the noise induced variances need to be accounted for in a similar fashion to the statistical representation in the reduce stage. First an absolute phase reference is created by assigned

a value for position A of $\mathbb{P}_A = 1$. One example path calculation, for Point C, is shown below to give a foundation for the final calculations. Since there are two paths from A to C, it is required to compute a noise-variance weighted average of the paths (Fried, 2013).

$$\frac{\mathbb{P}_C}{\mathbb{P}_A} = \frac{\sigma_{ADC}^2}{\sigma_{ABC}^2 + \sigma_{ADC}^2} u\Delta_{BA} u\Delta_{CB} + \frac{\sigma_{ABC}^2}{\sigma_{ABC}^2 + \sigma_{ADC}^2} u\Delta_{DA} u\Delta_{CD}$$

making note that the variances simply add along the path to get the corresponding path noise induced variance ($\sigma_{ABC}^2 = \sigma_{BA}^2 + \sigma_{CB}^2$). Carrying out the formulation for the remaining paths from reference point A to the remaining points, then solving in terms of the known variances and differential phasors leads to the final equations (Fried, 2013).

$$\begin{aligned}\mathbb{P}_C &= \frac{(\sigma_{DA}^2 + \sigma_{CD}^2)u\Delta_{BA}u\Delta_{CB} + (\sigma_{BA}^2 + \sigma_{CB}^2)u\Delta_{DA}u\Delta_{CD}}{\sigma_{ALL}^2} \\ \mathbb{P}_B &= \frac{(\sigma_{DA}^2 + \sigma_{CD}^2 + \sigma_{CB}^2)u\Delta_{BA} + \sigma_{BA}^2 u\Delta_{DA}u\Delta_{CD}(u\Delta_{CB})^*}{\sigma_{ALL}^2} \\ \mathbb{P}_D &= \frac{(\sigma_{BA}^2 + \sigma_{CB}^2 + \sigma_{CD}^2)u\Delta_{DA} + \sigma_{DA}^2 u\Delta_{BA}u\Delta_{CB}(u\Delta_{CD})^*}{\sigma_{ALL}^2}\end{aligned}$$

and

$$\begin{aligned}\sigma_A^2 &= \frac{(\sigma_{BA}^2 + \sigma_{CB}^2)(\sigma_{DA}^2 + \sigma_{CD}^2)}{2\sigma_{ALL}^2} \\ \sigma_B^2 &= \frac{(\sigma_{BA}^2 + \sigma_{CB}^2)(\sigma_{DA}^2 + \sigma_{CD}^2) + 2\sigma_{BA}^2(\sigma_{DA}^2 + \sigma_{CB}^2 + \sigma_{CD}^2)}{2\sigma_{ALL}^2} \\ \sigma_C^2 &= \frac{(\sigma_{BA}^2 + \sigma_{CB}^2)(\sigma_{DA}^2 + \sigma_{CD}^2)}{2\sigma_{ALL}^2} \\ \sigma_D^2 &= \frac{(\sigma_{BA}^2 + \sigma_{CB}^2)(\sigma_{DA}^2 + \sigma_{CD}^2) + 2\sigma_{DA}^2(\sigma_{BA}^2 + \sigma_{CB}^2 + \sigma_{CD}^2)}{2\sigma_{ALL}^2}\end{aligned}$$

where $\sigma_{ALL}^2 = \sigma_{BA}^2 + \sigma_{CB}^2 + \sigma_{CD}^2 + \sigma_{DA}^2$ (Fried, 2013). The end of the solve stage leaves the algorithm with four phasor values and their accompanying noise induced variances. The next stage uses these phasor values to calculate the other corresponding grid points.

c. *Rebuild Stage*

The rebuild stage of the NVWCER works in a similar, but reverse fashion, as the reduce stage. It recursively starts with a 2^{N+1} square grid of phasor values and uses

the corresponding differential phasor from the equivalent step of the reduce stage to calculate the a noise-variance weighted value for the phasors on a $2^{N+1}+1$ square grid, constantly repeating until the original lattice size is achieved. The algorithm first calculates the phasor at the center of each elemental square, then the middle phasor points on the edge of each one (Fried, 2013). The notations A, B, C, and D are used to refer to the calculated phasors from the previous step of the rebuild stage, or the output of the solve stage if it is the first iteration. The use of $x_{\#}$ and $y_{\#}$ are used to represent the differential phasors from the appropriate step of the reduce stage, with the same amount of current grid points. The equation for the phasor, u , and its noise induced variance, σ_u^2 , at the center of the elemental square follow the notation shown in Figure 13, where the solid circle inside an open circle represents the phasor to be calculated

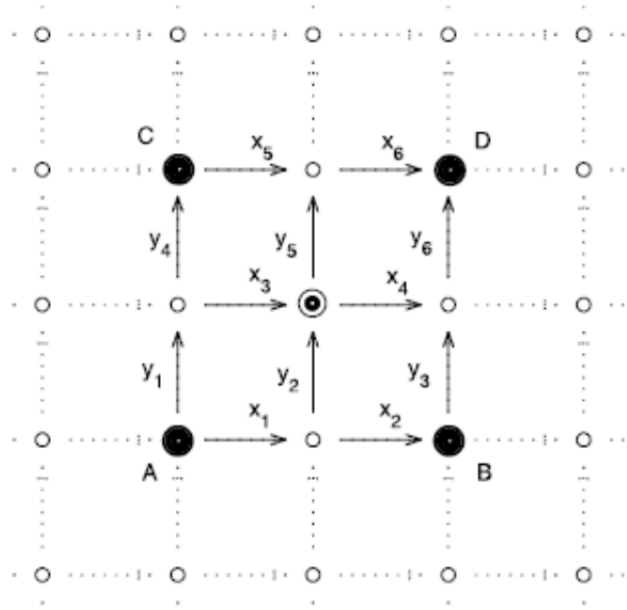


Figure 13. Calculation for center phasor of each elemental square in Rebuild stage (From Fried 2001).

$$u = \left\{ \left[\frac{A \left\{ \left(\frac{x_1 y_2}{(\sigma_{x_1}^2 + \sigma_{y_2}^2)} + \frac{y_1 x_3}{\sigma_{y_1}^2 + \sigma_{x_3}^2} \right) \sigma_{Axy}^2 \right\}}{\sigma_A^2 + \sigma_{Axy}^2} + \frac{B \left\{ \left(\frac{x_2^* y_2}{(\sigma_{x_2}^2 + \sigma_{y_2}^2)} + \frac{y_3 x_4^*}{\sigma_{y_3}^2 + \sigma_{x_4}^2} \right) \sigma_{Bxy}^2 \right\}}{\sigma_B^2 + \sigma_{Bxy}^2} + \frac{C \left\{ \left(\frac{y_4^* x_3}{(\sigma_{y_4}^2 + \sigma_{x_3}^2)} + \frac{x_5 y_5^*}{\sigma_{x_5}^2 + \sigma_{y_5}^2} \right) \sigma_{Cxy}^2 \right\}}{\sigma_C^2 + \sigma_{Cxy}^2} + \frac{D \left\{ \left(\frac{x_6^* y_5^*}{(\sigma_{x_6}^2 + \sigma_{y_5}^2)} + \frac{y_6^* x_4^*}{\sigma_{y_6}^2 + \sigma_{x_4}^2} \right) \sigma_{Dxy}^2 \right\}}{\sigma_D^2 + \sigma_{Dxy}^2} \right] \sigma_u^2 \right\}$$

and

$$\sigma_u^2 = \frac{1}{\frac{1}{\sigma_A^2 + \sigma_{Axy}^2} + \frac{1}{\sigma_B^2 + \sigma_{Bxy}^2} + \frac{1}{\sigma_C^2 + \sigma_{Cxy}^2} + \frac{1}{\sigma_D^2 + \sigma_{Dxy}^2}}$$

where

$$\begin{aligned} \sigma_{Axy}^2 &= \frac{1}{\frac{1}{\sigma_{x_1}^2 + \sigma_{y_2}^2} + \frac{1}{\sigma_{y_1}^2 + \sigma_{x_3}^2}} \\ \sigma_{Bxy}^2 &= \frac{1}{\frac{1}{\sigma_{x_2}^2 + \sigma_{y_2}^2} + \frac{1}{\sigma_{y_3}^2 + \sigma_{x_4}^2}} \\ \sigma_{Cxy}^2 &= \frac{1}{\frac{1}{\sigma_{x_3}^2 + \sigma_{y_4}^2} + \frac{1}{\sigma_{y_5}^2 + \sigma_{x_5}^2}} \\ \sigma_{Dxy}^2 &= \frac{1}{\frac{1}{\sigma_{x_6}^2 + \sigma_{y_5}^2} + \frac{1}{\sigma_{y_6}^2 + \sigma_{x_4}^2}} \end{aligned}$$

Once the phasors at the center of all the unit squares are calculated, the next step is to calculate the phasors at the middle of the edges of the unit squares. When the edge of the unit square lies on the interior of the lattice then the following equations are used, representing the notation in Figure 14. The sold black circles represent phasors from the previous steps of the rebuild stage, the small dots with open circles around them represent the calculated phasors at the center of each elemental square, and the small black circles are the phasors to be calculated.

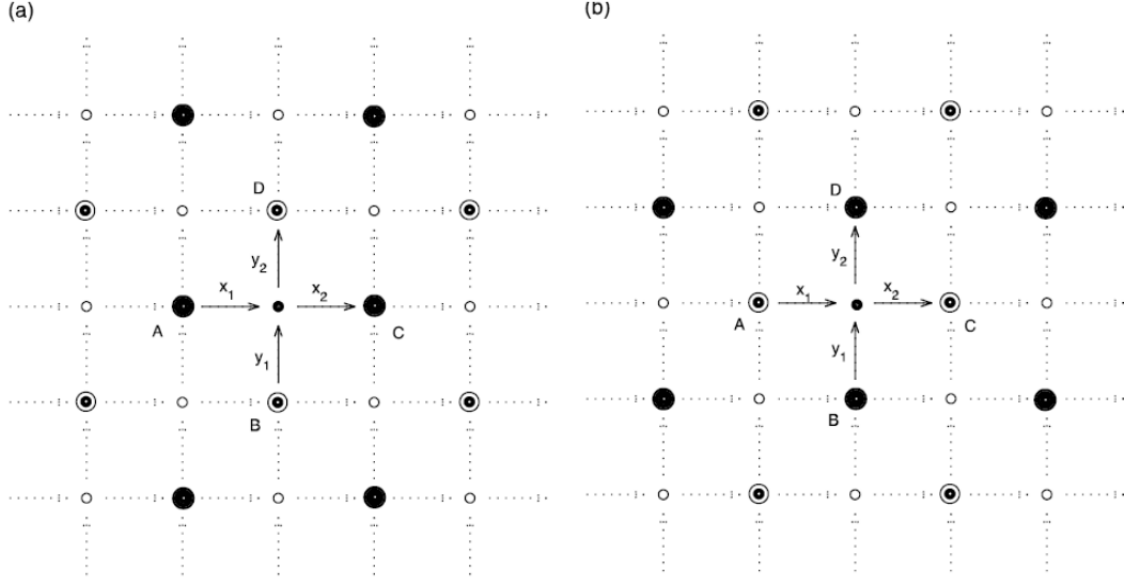


Figure 14. Calculation for Phasors at the middle of edges of elementary squares, located on interior of lattice (From Fried 2001)

The calculations apply to both setups in Figure 14, either edge on top or bottom of elemental square (left) or on the left or right of elemental square (right).

$$u = \left\{ \left(\frac{Ax_1}{\sigma_A^2 + \sigma_{x_1}^2} + \frac{By_1}{\sigma_B^2 + \sigma_{y_1}^2} + \frac{Cx_2^*}{\sigma_C^2 + \sigma_{x_2}^2} + \frac{Dy_2^*}{\sigma_D^2 + \sigma_{y_2}^2} \right) \sigma_u^2 \right\}$$

and

$$\sigma_u^2 = \frac{1}{\frac{1}{\sigma_A^2 + \sigma_{x_1}^2} + \frac{1}{\sigma_B^2 + \sigma_{y_1}^2} + \frac{1}{\sigma_C^2 + \sigma_{x_2}^2} + \frac{1}{\sigma_D^2 + \sigma_{y_2}^2}}$$

If the phasor is located on the edge of the lattice then there are four possibilities; either on the sides of the lattice or on the top or bottom of the lattice. Figure 15 shows the notation for all four possibilities, and the same symbols apply from Figure 14.

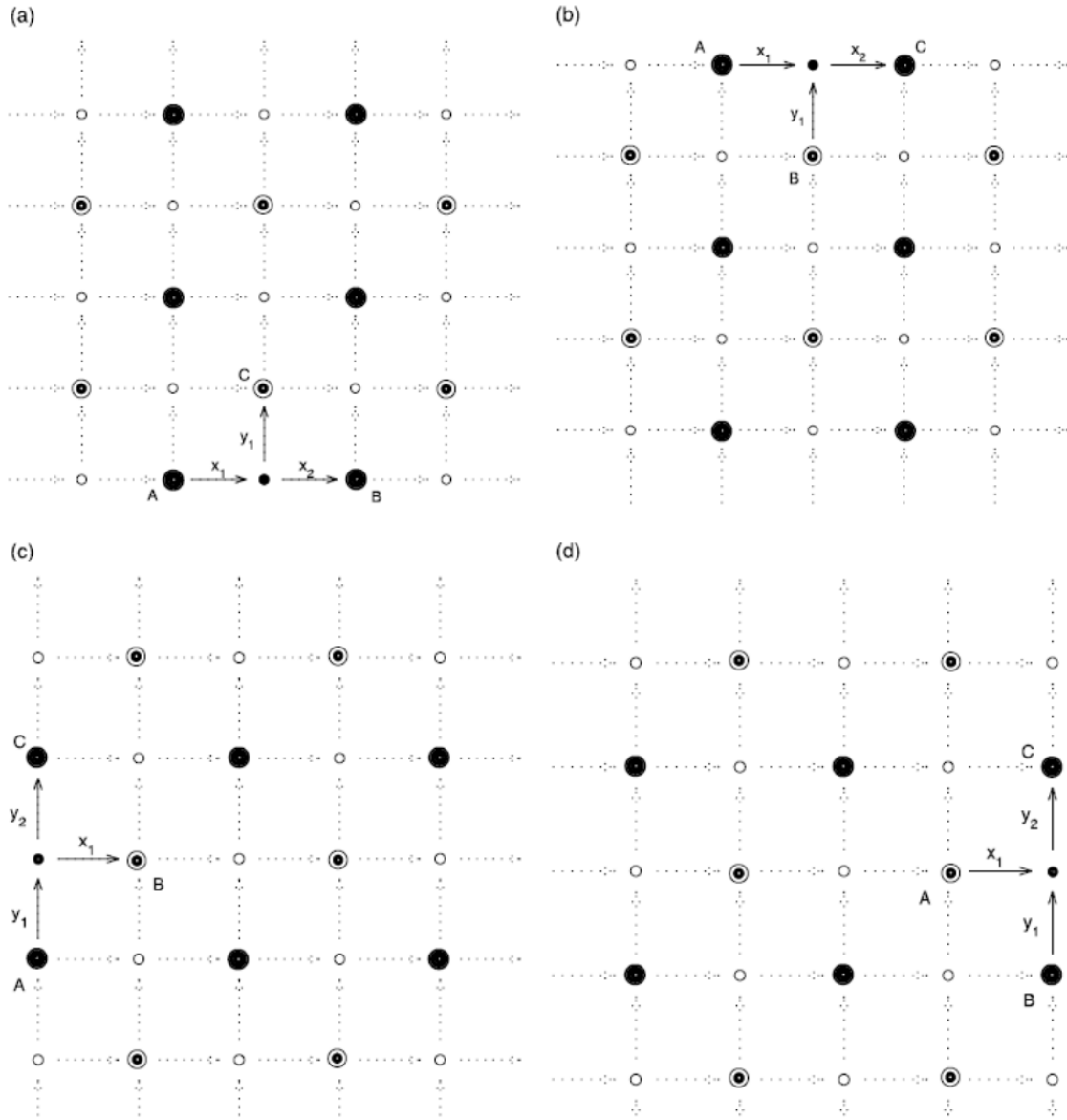


Figure 15. Phasor calculation setup for Rebuild stage; phasor located on edge of lattice
(From Fried 2001)

Dependent on where the phasor is located differing set of equations apply.

For a phasor located on the bottom of the lattice, show in Figure 15.a,

$$u = \left\{ \left(\frac{Ax_1}{\sigma_A^2 + \sigma_{x_1}^2} + \frac{Bx_2^*}{\sigma_B^2 + \sigma_{x_2}^2} + \frac{Cy_1^*}{\sigma_C^2 + \sigma_{y_1}^2} \right) \sigma_u^2 \right\}$$

$$\sigma_u^2 = \frac{1}{\frac{1}{\sigma_A^2 + \sigma_{x_1}^2} + \frac{1}{\sigma_B^2 + \sigma_{x_2}^2} + \frac{1}{\sigma_C^2 + \sigma_{y_1}^2}}$$

For phasors located on the top as show in Figure 15.b;

$$u = \left\{ \left(\frac{Ax_1}{\sigma_A^2 + \sigma_{x_1}^2} + \frac{By_1}{\sigma_B^2 + \sigma_{y_1}^2} + \frac{Cx_2^*}{\sigma_C^2 + \sigma_{x_2}^2} \right) \sigma_u^2 \right\}$$

$$\sigma_u^2 = \frac{1}{\frac{1}{\sigma_A^2 + \sigma_{x_1}^2} + \frac{1}{\sigma_B^2 + \sigma_{y_1}^2} + \frac{1}{\sigma_C^2 + \sigma_{x_2}^2}}$$

For phasors located on the left edge as shown in Figure 15.c;

$$u = \left\{ \left(\frac{Ay_1}{\sigma_A^2 + \sigma_{y_1}^2} + \frac{Bx_1^*}{\sigma_B^2 + \sigma_{x_1}^2} + \frac{Cy_2^*}{\sigma_C^2 + \sigma_{y_2}^2} \right) \sigma_u^2 \right\}$$

$$\sigma_u^2 = \frac{1}{\frac{1}{\sigma_A^2 + \sigma_{y_1}^2} + \frac{1}{\sigma_B^2 + \sigma_{x_1}^2} + \frac{1}{\sigma_C^2 + \sigma_{y_2}^2}}$$

For phasors located on the right edge as show in Figure 15.d;

$$u = \left\{ \left(\frac{Ax_1}{\sigma_A^2 + \sigma_{x_1}^2} + \frac{By_1}{\sigma_B^2 + \sigma_{y_1}^2} + \frac{Cy_2^*}{\sigma_C^2 + \sigma_{y_1}^2} \right) \sigma_u^2 \right\}$$

$$\sigma_u^2 = \frac{1}{\frac{1}{\sigma_A^2 + \sigma_{x_1}^2} + \frac{1}{\sigma_B^2 + \sigma_{y_1}^2} + \frac{1}{\sigma_C^2 + \sigma_{y_2}^2}}$$

With the three stages presented, along with the supporting equations and notation, it is possible to carry out the full extent of the NVWCER. However, this algorithm is only partially complete, because it has done nothing for unwrapping or branch point placement. This is purely a calculation of the phase in the complex domain, with no consideration of branch points yet. A second part of the algorithm will take the complex phasors attempt to identify and place the branch points, as well as provide an unwrapped and continuous phase in the real domain.

d. Phase Unwrapping and Branch Point Analysis

Give the output of phasors from the rebuild stage of the NVWCER, there are two main issues. First it is in the complex domain as phasors and not phases, which are required for an AO system. Secondly, the system has not accounted for branch points, in which there will be large 2π discontinuities. This “smoothing” algorithm, presented by Dr. Fried, takes the phasor data and extracts a continuous, unwrapped phase function that has ideally placed branch points close together and in areas of low intensity as to minimize the effects on the AO system.

The algorithm first starts by calculating the differential phasors from the output phasors of the rebuild stage; these ideally are the same measurements as the initial stage, however with noise considerations taken (Fried, 2001). These differential phasors are summed up around in each unit circle to identify any branch points. The next step uses the location of the branch points to calculate the hidden phase, and remove that from the phasors, ideally leaving a scalar, branch point free phase function. Using the scalar phase function, which ideally has no branch points, the phase can be calculated by simply setting a reference zero phase point and just adding the differential phasors to calculate each phasor. Finally, the total phase function is created as a sum of the hidden phase and scalar phase function, which has accounted for branch points and placed them preferably in locations of low intensity.

First, the phase differences are calculated across the entire lattice according to

$$\begin{aligned} x\Delta\phi_{i,j} &= \arg(u_{i+1,j}u_{i,j}^*) \\ y\Delta\phi_{i,j} &= \arg(u_{i,j}u_{i,j+1}^*) \end{aligned}$$

and then the curl, $C(i,j)$, is calculated about each unit square, or sub-aperture, according to

$$C(i,j) = x\Delta\phi_{i,j} + y\Delta\phi_{i+1,j} - x\Delta\phi_{i,j+1} - y\Delta\phi_{i,j}$$

Once the curls are calculated then the branch points can be located. Assuming the range of the phases is from $-\pi$ to $+\pi$, then that is used for the reference of finding branch points. A positive branch point is denoted by $P = \{(i_p, j_p)\}$ and negative branch points by $N = \{(i_n, j_n)\}$ (Fried, 2001). The criteria for being classified a branch points is as follows:

$$\begin{aligned} (i_p, j_p) &\in P \text{ if } C(i_p, j_p) > +\pi \\ (i_p, j_p) &\notin P \text{ if } C(i_p, j_p) \leq +\pi \\ (i_n, j_n) &\in P \text{ if } C(i_n, j_n) < -\pi \\ (i_n, j_n) &\notin P \text{ if } C(i_n, j_n) \geq -\pi \end{aligned}$$

Since the indexing for each curl function, $C(i,j)$, is indexed off of the bottom left corner of each unit square, then it is necessary to account for the location of

the branch point being located at the center. The actual locations of the branch points are denoted $\mathbf{p}_p = (\xi_{i_p}, \eta_{j_p})$ and $\mathbf{p}_n = (\xi_{i_n}, \eta_{j_n})$, where

$$\begin{aligned}\xi_{i_p} &= \frac{1}{2}(x_{i_p} + x_{i_{p+1}}), & \eta_{j_p} &= \frac{1}{2}(y_{j_p} + y_{j_{p+1}}) \\ \xi_{i_n} &= \frac{1}{2}(x_{i_n} + x_{i_{n+1}}), & \eta_{j_n} &= \frac{1}{2}(y_{j_n} + y_{j_{n+1}}).\end{aligned}$$

Once the curls around each unit square are calculated, and all of the branch points are detected, the total amount of branch points is tallied. If there are more positive (negative) branch points then negative (positive), an artificial negative (positive) branch point is placed at the maximum (minimum) of the potential field, $V(i,j)$. This process is repeated until the number of positive and negative branch points are equal.

$$V(i,j) = \sum_{(i_p, j_p) \in P} \frac{1}{\sqrt{(x_i - \xi_{i_p})^2 + (y_j - \eta_{j_p})^2}} - \sum_{(i_n, j_n) \in N} \frac{1}{\sqrt{(x_i - \xi_{i_n})^2 + (y_j - \eta_{j_n})^2}}$$

Then the hidden phase is calculated and removed from the original phasor lattice to create a scalar function, $\tilde{u}_{i,j}$, free of branch points.

$$\begin{aligned}\phi_{hid}(i,j) &= \arg \left(\frac{\prod_{(i_p, j_p) \in P} (x_i - \xi_{i_p}) + i(y_j - \eta_{j_p})}{\sum_{(i_n, j_n) \in N} (x_i - \xi_{i_n}) + i(y_j - \eta_{j_n})} \right) \\ \tilde{u}_{i,j} &= \frac{u_{i,j}}{\exp(i\phi_{hid}(i,j))}\end{aligned}$$

The scalar function is free of branch points, so the phase can be calculated through a simple summation through the paths of differentials. When adding up the scalar phase differences, a reference value of 0 is set at an arbitrary point.

$$\begin{aligned}\widetilde{x\Delta\phi}_{i,j} &= \arg(\tilde{u}_{i+1,j}\tilde{u}_{i,j}^*) \\ \widetilde{y\Delta\phi}_{i,j} &= \arg(\tilde{u}_{i,j+1}\tilde{u}_{i,j}^*) \\ \phi_{scalar}(i,j) &= \sum_{i'=1}^{i'-1} \widetilde{x\Delta\phi}_{i',j} + \sum_{j'=1}^{j'-1} \widetilde{y\Delta\phi}_{i,j'}\end{aligned}$$

With the scalar phase and hidden phase calculated, the total phase is calculated as a summation of the hidden and scalar phase functions.

$$\phi(i,j) = \phi_{hid}(i,j) + \phi_{scalar}(i,j)$$

This process creates a phase which, as best as possible, is unwrapped with branch points placed in areas of lowest optical intensity. It is important to note again that

this process is defined for Hudgin geometry. The next section discusses how to properly adapt this algorithm for a fried-based geometry, making it compatible with a SHWFS.

e. Adaptation to Fried Geometry

Previously mentioned, the SHFWS operates on either Fried or Southwell geometry, but the NVWCER operates on Hudgin geometry. The simplest way is to convert the SHFWS data into two interleaved Hudgin lattices, carry out the NVWCER process on each of them, and then interleave them back together. Looking at a basic Hudgin lattice, as shown in Figure 16, the black dots represent the points on one Hudgin lattice, and the white dots, the other lattice.

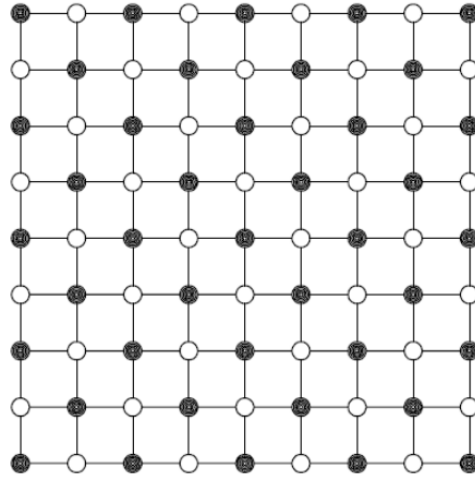


Figure 16. Fried-to-Hudgin Geometry Conversion (From Fried 2001)

The phase differences in the corresponding Hudgin lattices show in Figure 17 have phase differences which are the difference of diagonally opposite phases in a given supaperture. In Fried geometry, the x or y slope measurements of a SHWFS correspond to the difference of the average of the two phases on each side, for x, or top and bottom for y, divided by the distance (Fried, 2001). Mathematically, the diagonal phase differences, as shown in the Hudgin lattices, correspond to the sum or difference of those same x and y slope measurements from the SHWFS for that given supaperature (Fried, 2001).

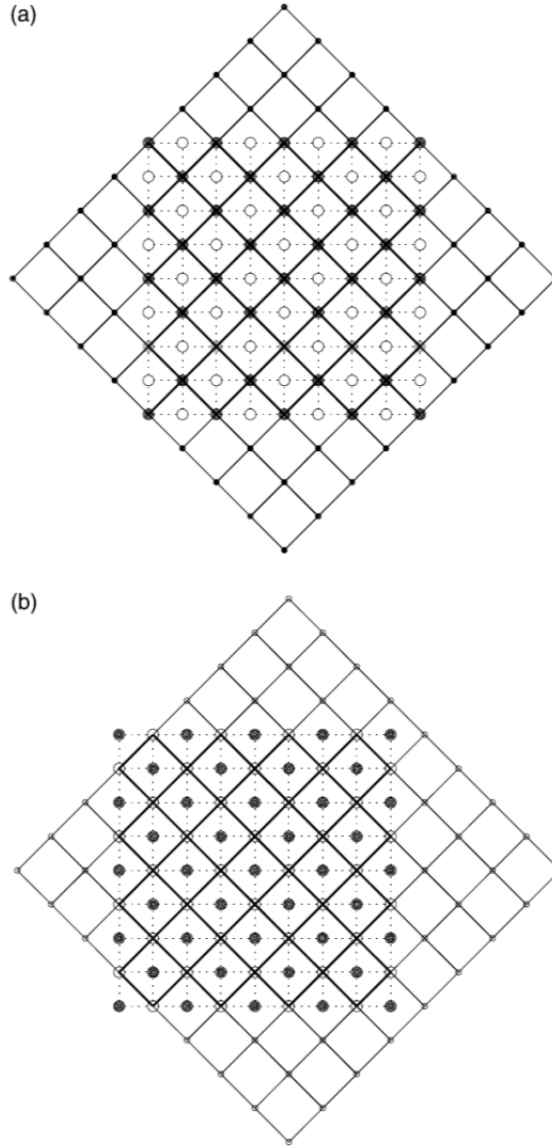


Figure 17. Corresponding Hudgin Geometries for a SHFWS Fried Geometry (From Fried 2001)

Given both of the lattices in Figure 17, values not corresponding to the original lattice are given a phasor value of 0 and a variance value of infinity, this in essence allows the NVWCER to operate without actually using them in subsequent calculations. Once the NVWCER is ran on both lattices the phase points are then inserted back into the original positions to form a Fried-geometry lattice. Since there is no absolute reference phase point, only a relative reference phase for each individual lattice,

there is a chance they are out of phase with each other. In order to bring them back in phase, the phases corresponding to the white dots in Figure 16 are compared with an average of the four adjacent grid points to calculate the average phase shift. This phase shift is then applied to the lattice.

First the noise-variance weighted average of the four adjacent grid points to each white grid point from Figure 16 is calculated. The phases of the un-synced lattice are referred to as $\tilde{u}(i, j)$.

$$\tilde{u}(i, j) = \frac{\frac{\tilde{u}(i-1, j)}{\sigma^2(i-1, j)} + \frac{\tilde{u}(i+1, j)}{\sigma^2(i+1, j)} + \frac{\tilde{u}(i, j-1)}{\sigma^2(i, j-1)} + \frac{\tilde{u}(i, j+1)}{\sigma^2(i, j+1)}}{\frac{4}{\tilde{\sigma}^2(i, j)}}$$

for $i + j$ odd

where

$$\tilde{\sigma}^2(i, j) = \frac{4}{\frac{1}{\sigma^2(i-1, j)} + \frac{1}{\sigma^2(i+1, j)} + \frac{1}{\sigma^2(i, j-1)} + \frac{1}{\sigma^2(i, j+1)}}$$

Then the weighted sum of the products of the averaged value, $\tilde{u}(i, j)$, and the conjugate of the original un-synced phases, $\tilde{u}(i, j)$, is found to calculate the weighted average phase shift (Fried, 2001).

$$\tilde{\phi}(i, j) = \arg\left\{ \sum_{i+j \text{ odd}} \frac{\tilde{u}(i, j) \tilde{u}^*(i, j)}{\tilde{\sigma}^2(i, j)} \right\}$$

$$\tilde{\sigma}^2(i, j) = \tilde{\sigma}^2(i, j) + \sigma^2(i, j) \quad , \quad \text{for } i + j \text{ odd}$$

Once the average phase shift is applied the final phasor values, $u(i, j)$, are calculated and the phase shift is applied to correct Hudgin lattice points.

$$u(i, j) = \begin{cases} \tilde{u}(i, j) & \text{for } i + j \text{ odd} \\ \exp(i\tilde{\phi}) \tilde{u}(i, j) & \text{for } i + j \text{ even} \end{cases}$$

This final phasor lattice can then be applied to the branch point analysis and unwrapping algorithm. It is important to note that due to the interleaving of two lattices, and a finite estimate of the phase shift, there is subsequent error that can

propagate forward as a result. The attempt to re-sync the phases can lead to minor “egg-crating” effects due to areas of minor phase shift discrepancies (Fried, 2001)

V. ANALYSIS

A MATLAB simulation environment is created, to analyze the different reconstruction methods with varying Rytov variances, SNR, and sensor grid size. It is important to note when random noise is added to generate the desired SNR, the corresponding SHFWS output is simultaneously sent to all three reconstructors. This assures comparison is made using the exact same sensor data. The principal measurement for the quality of reconstruction is known as the Strehl ratio. It is a weighted measurement of the difference between original phase and reconstructed phase at each point. The Strehl ratio is described by the following equation,

$$S = \frac{|\left[\sum_{i=1}^N \sum_{j=1}^N u_{original}(i,j) u_{rec}^*(i,j)\right]|^2}{\left[\sum_{i=1}^N \sum_{j=1}^N |u_{original}(i,j)|\right]^2}$$

where $u_{original}$ is the complex function of the original phase and amplitude sampled at the reconstructed points and u_{rec} is the complex function of the reconstructed phase with unity amplitude.

$$\begin{aligned} u_{original}(i,j) &= A(i,j)e^{i\phi(i,j)} \\ u_{rec}(i,j) &= e^{i\phi_{rec}(i,j)} \end{aligned}$$

Looking at the equation for the Strehl ratio, where the value of the original is multiplied by the complex conjugate of the reconstructed that, it is essentially calculating the difference in phase. The closer the values are, the closer the values of the exponential are driven to zero, leading to a value of one.

All reconstructors are evaluated using all possible combinations of the following factors: SNR (10, 60, and 200), Number of sensors along one side (16, 32, and 64), and Rytov variances (0.109, 0.349, and 0.567). In each section of the analysis only certain results are discussed, in order to obtain an accurate analysis. A table of results for all combinations is included as well

A. SOURCE PHASE AND INTENSITY DATA

The data used in testing and analysis of the various reconstruction algorithms was generated using a program known as WaveProp. This is a proprietary MATLAB program

which simulates the propagation of a beam through a turbulent atmosphere, with given parameters like distance, turbulence, wavelength, and size. The data used in this simulation had the parameters of a propagation distance of 4km, a wavelength of one micron (1 μ m), and a sensor side length of 0.3m. The input for WaveProp is C_n^2 , but given the path length and wavelength, the equation for Rytov variance can be used to calculate the proper C_n^2 for a desired Rytov number.

$$\sigma_R^2 = 1.23 C_n^2 \left(\frac{2\pi}{\lambda} \right)^{\frac{7}{6}} L^{\frac{11}{6}}$$

The following are images of the phase and amplitude data.

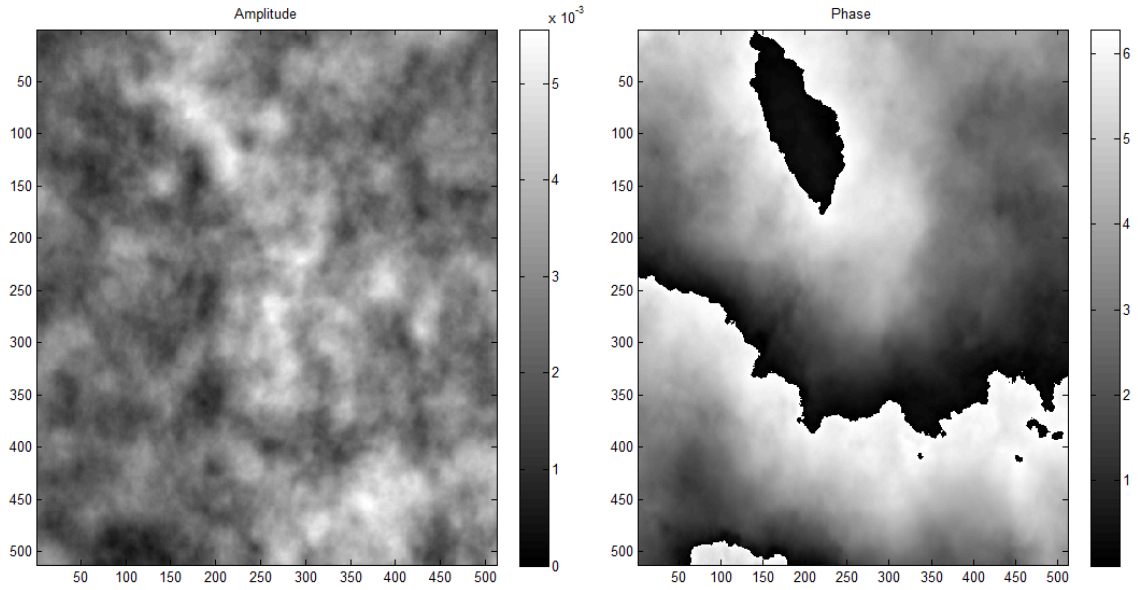


Figure 18. Simulation data for Rytov Variance of 0.109

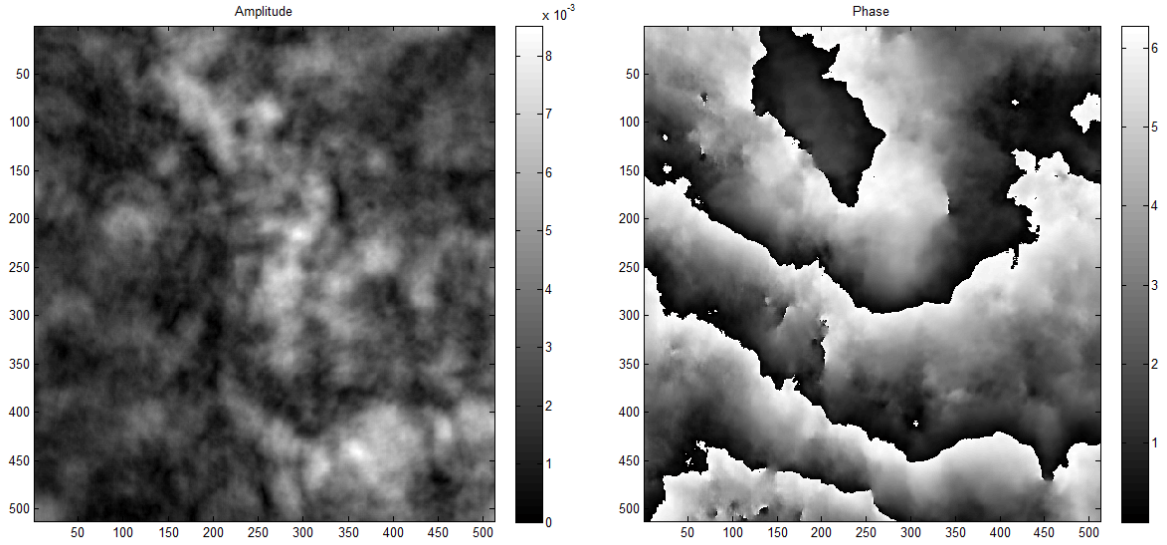


Figure 19. Simulation data for Rytov Variance of 0.349

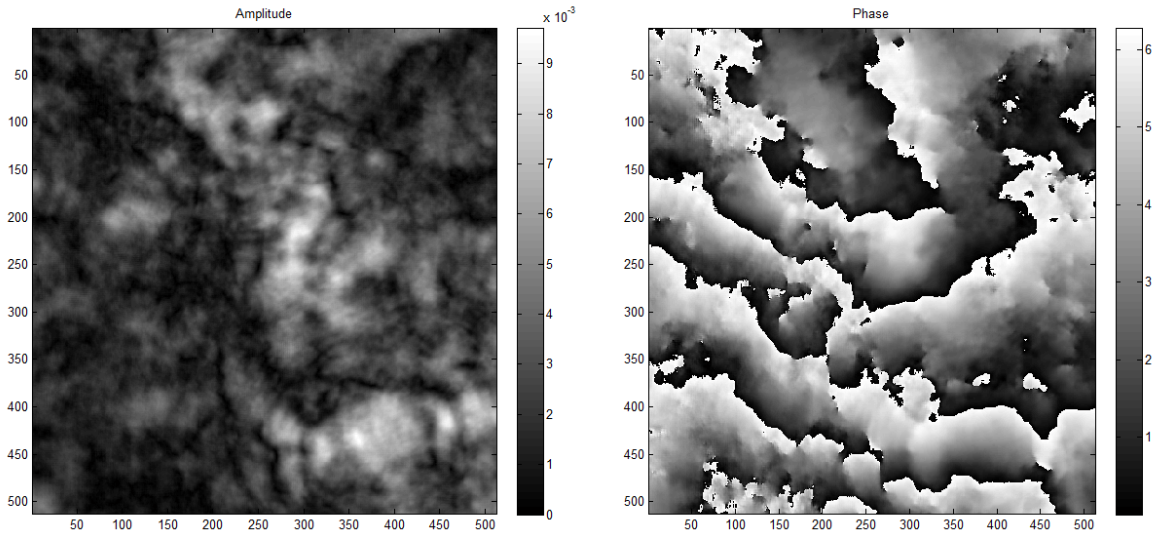


Figure 20. Simulation data for Rytov Variance of 0.567

The phase data presented is wrapped, meaning it is on the domain of zero to 2π . The interest of reconstruction is generating an unwrapped phase, meaning it exist on the domain of $-\infty$ to $+\infty$, so that it is as close to a continuous as possible. This relates to the need for generating commands for the deformable mirror, which is continuous as well

and cannot create step discontinuities. When results are presented both the wrapped and unwrapped phase will be shown, however in the Strehl ratio calculation it has no effect due to the domain a complex exponential repeating every 2π .

B. INITIAL TEST

1. Variation in Number of Sensors

The number of sensors is evaluated at 16, 32, and 64 along one for each reconstructor. The data presented in this section for analysis has an SNR of 60, and is analyzed for a Rytov variance 0.567. The primary concern is to see if there is a limit to the number of sensors needed for a relatively high Rytov Variance.

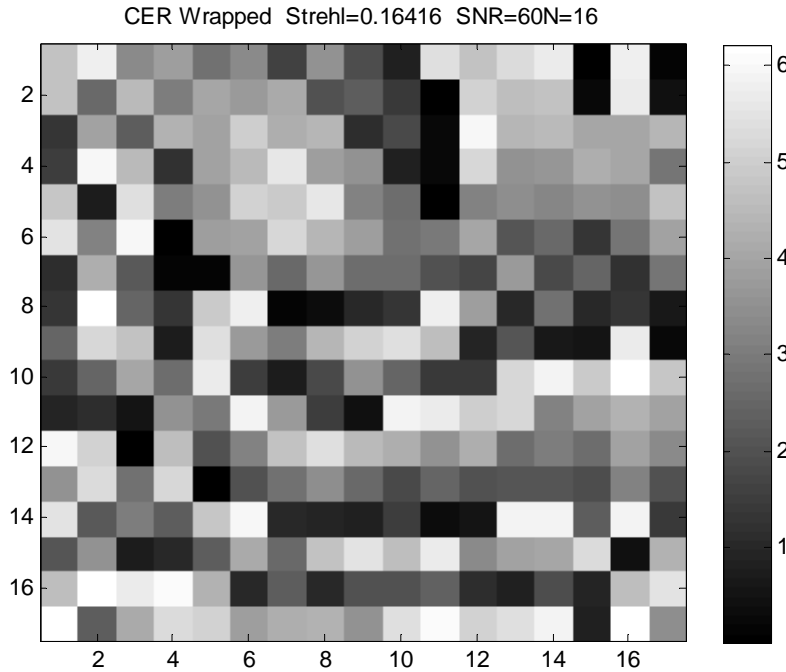


Figure 21. NVWCER Reconstruction. $S=0.146$, $\text{SNR}=60$, and $N=16$

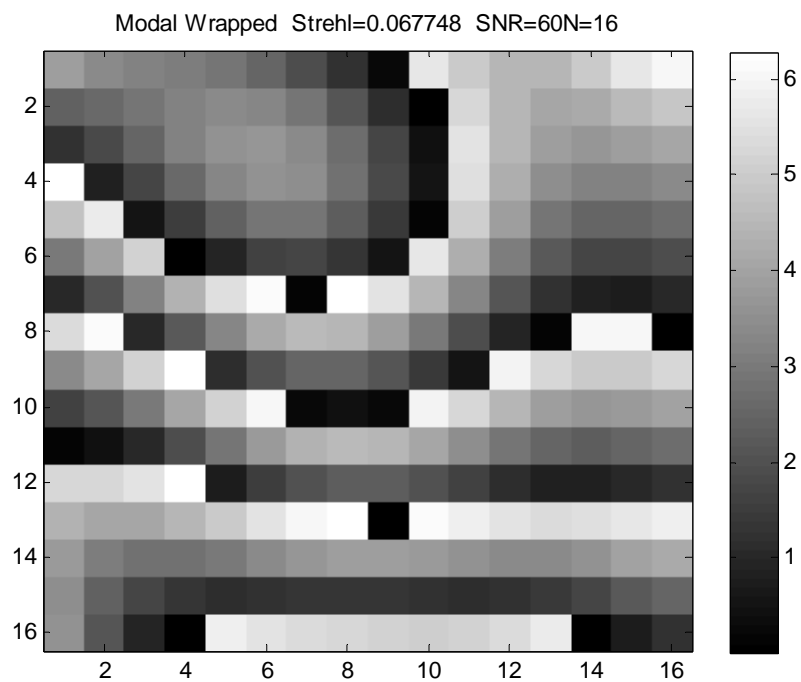


Figure 22. Modal Reconstruction. $S=0.068$, $\text{SNR}=60$, and $N=16$

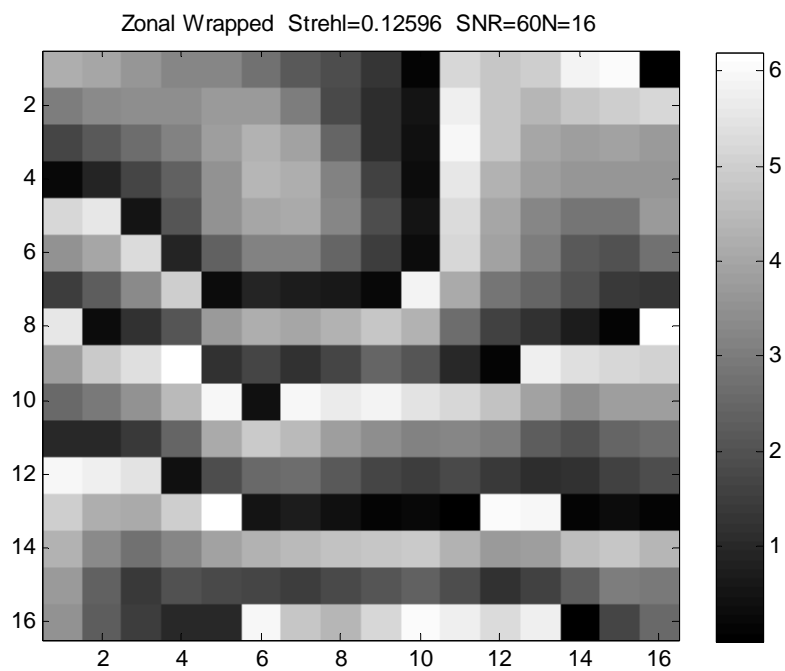


Figure 23. Zonal Reconstruction. $S=0.126$, $\text{SNR}=60$, and $N=16$

The reconstruction of a high Rytov variance of 0.567 is difficult with a small amount of sensors. The Modal reconstruction performs the worst with a Strehl of 0.068, which is a combination of the smoothing that is inherent to the modal reconstruction process and a low sensor count. The Zonal and NVWCER performed nearly identical, with Strehl's of 0.126 and 0.146, respectively. Zonal reconstruction is a method proven to work well with lower disturbance regimes, and given a lower sensor count the NVWCER has some difficulties properly identify the branch point locations. The following set of data is for 32 sensors along a side.

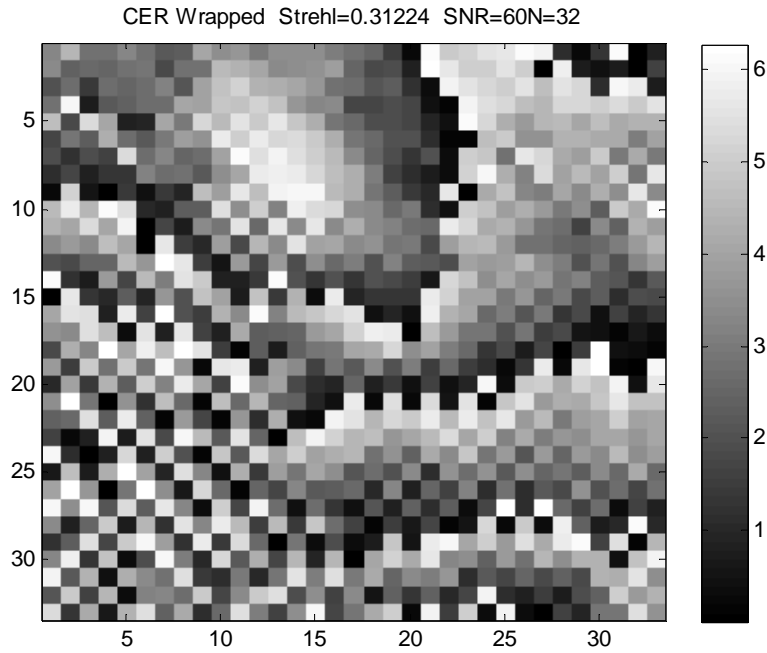


Figure 24. NVWCER Reconstruction. $S=0.312$, $SNR=60$, $N=32$

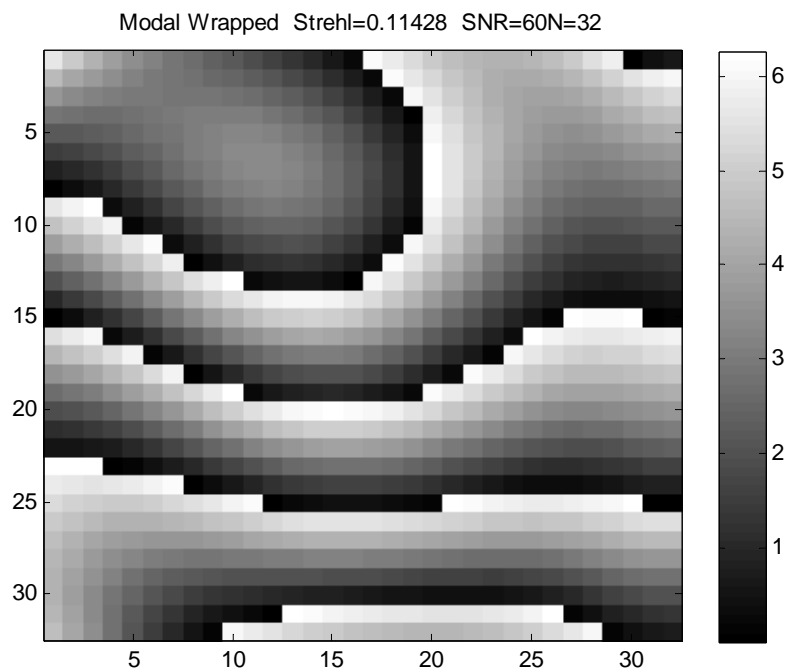


Figure 25. Modal reconstruction. $S=0.114$, $\text{SNR}=60$, and $N=32$

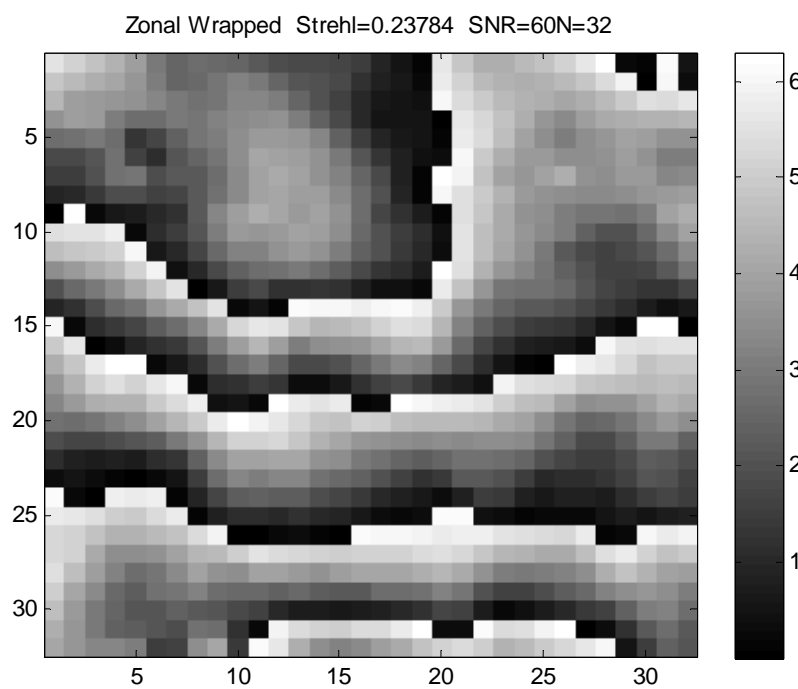


Figure 26. Zonal Reconstruction. $S=0.238$, $\text{SNR}=60$, and $N=32$

When the number of sensors is increased by a factor of 2, to 32, the Strehl increases for all three reconstruction methods as anticipated. The Modal again performs the worst of all three reconstructors. The NVWCER outperforms the Zonal reconstructor by a larger margin this time, with Strehl's of 0.312 and 0.238, respectively. Now that the NVWCER is receiving adequate data, it is able to perform better in the presence of high turbulence, as designed, than other methods. It is also important to note that in Figure 24, the checkerboard pattern is beginning to emerge, which was attributed to the process of the NVWCER for a SHFWS design needing to bring the two lattices into phase with each other. The last set of data is for a grid of 64 sensors along a side.

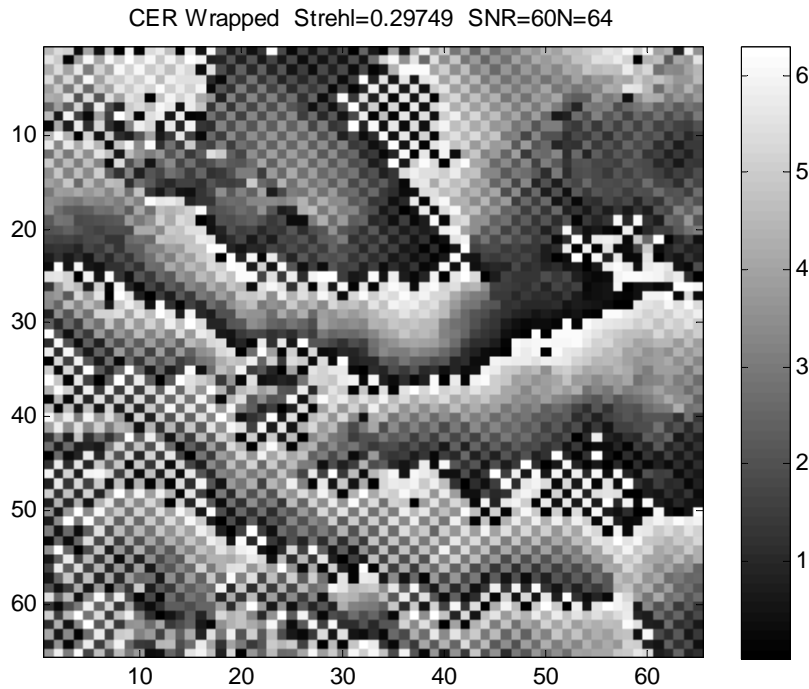


Figure 27. NVWCER Reconstruction. $S=0.297$, $SNR=60$, and $N=64$

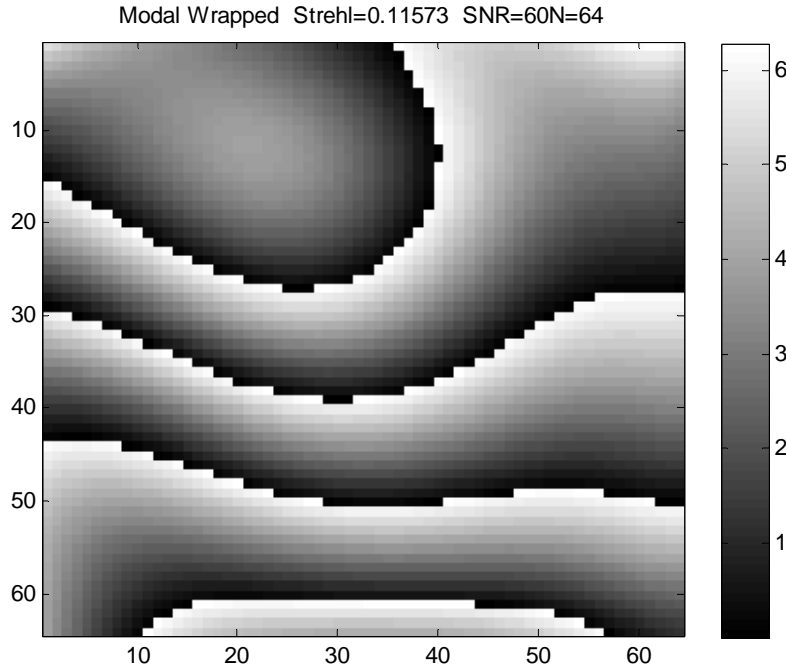


Figure 28. Modal Reconstruction. $S=0.116$, $\text{SNR}=60$, and $N=64$

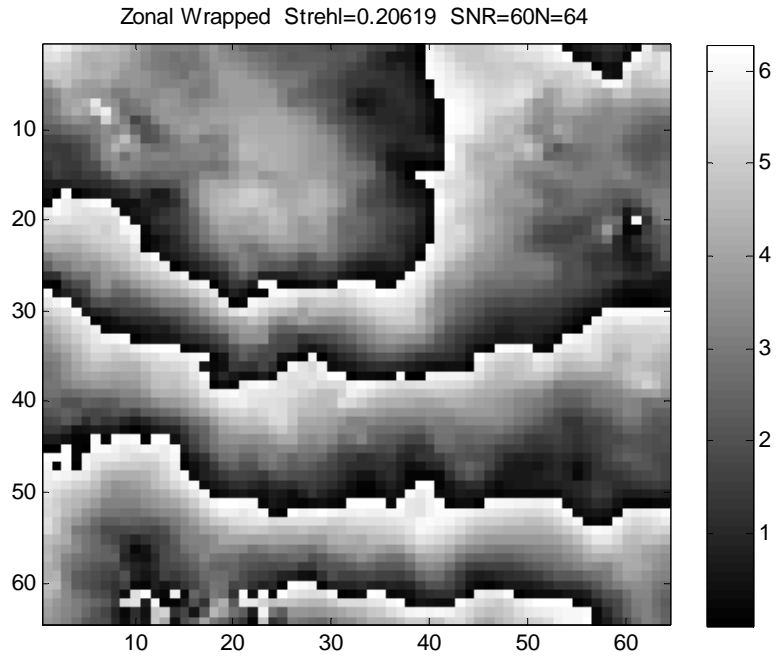


Figure 29. Zonal Reconstruction. $S=0.206$, $\text{SNR}=60$, and $N=64$

The reconstructors all perform in the same manner as the previous two sections with modal performing the worst and NVWCER performed the best. However, there was

a decrease in the Strehl ratio of both the NVWCER and zonal reconstructor of 0.2 and 0.3, respectively, which would lead one to believe that it is possible to over sample the data. Possible outlying factors could also be the location density of the random noise added to the system. While the SHWFS data is sent to all three reconstructors simultaneously for any trial, if any of the factors are changed a new random matrix of noise data is generated; the SNR is on a measurement defined over the entirety of the image.

The variation in the number of sensors has shown a trend that increasing the number of sensors to a threshold will have a positive correlation on the reconstruction process and the Strehl ratio. The NVWCER has shown it requires a minimum amount of sensors to outperform the zonal reconstructor to a significant degree, which is attributed to the fact it needs enough information in order to locate and handle the branch points. The next section will analyze the variation of SNR.

2. Variation in SNR

The SNR is used to represent the relative capabilities of the sensor itself. Depending on the SHWFS sensitivity the SNR can vary to a large degree. An SNR of 10 is considered relatively low, while a value of 60 is more realistic, and 200 is a sensor susceptible to very little noise. The reconstructors will be tested with all three SNR's. A Rytov variance of 0.349 will be used and the number of sensors will be set to 32 on a side.

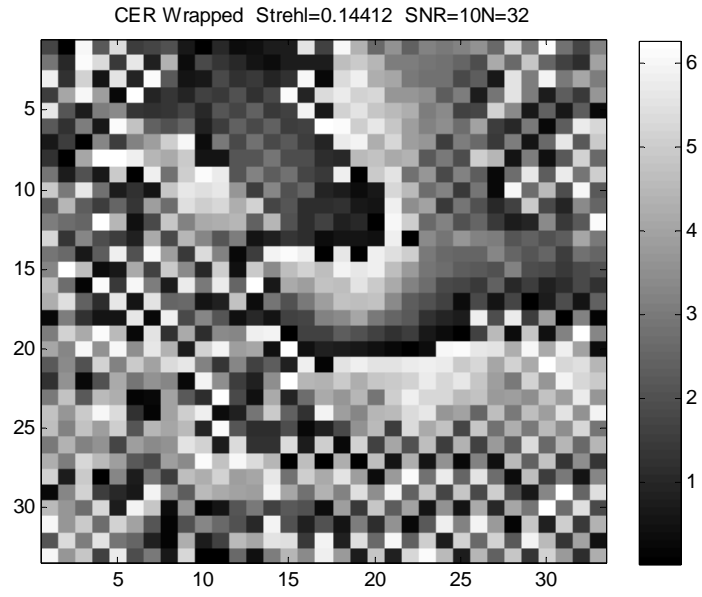


Figure 30. NVWCER Reconstructor. $S=0.144$, $\text{SNR}=10$, $N=32$

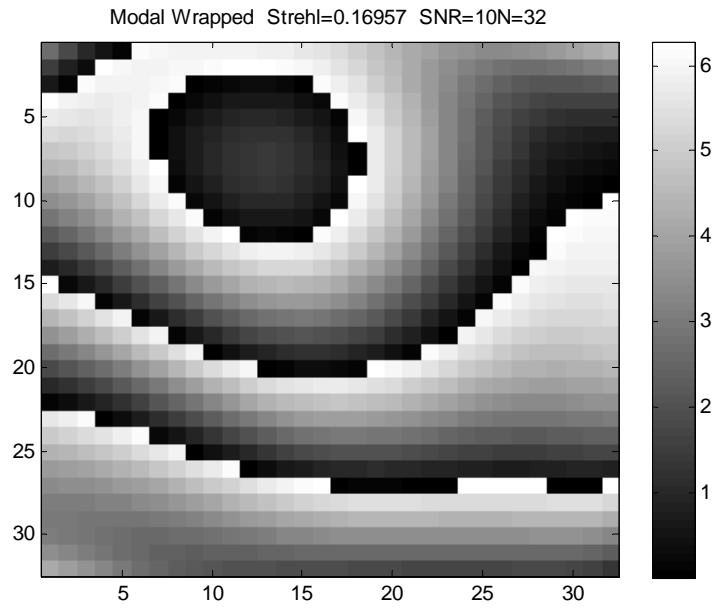


Figure 31. Modal Reconstruction. $S=0.169$, $\text{SNR}=10$, and $N=32$

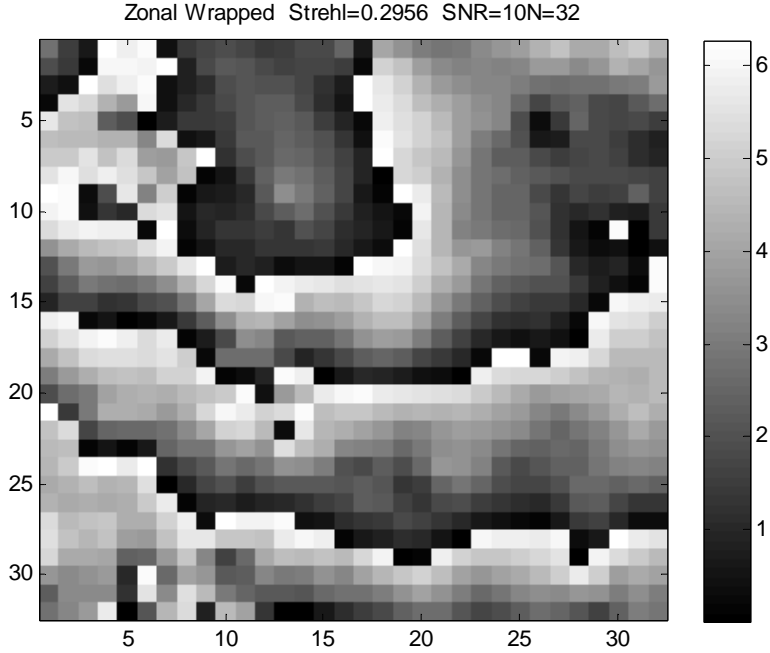


Figure 32. Zonal Reconstruction. $S=0.296$, $SNR=10$, and $N=32$

With such a low SNR all of the reconstructors obtain a relatively low Strehl ratio. The zonal reconstructor performs the best, with a value of 0.296, the modal process with a Strehl of 0.169, and NVWCER with a Strehl of 0.144. The modal reconstructors inherent smoothing assisted with the high amount of noise. The NVWCER is designed to handle noisy measurements however, with an SNR of 10, all paths of a high amount of noise. The noise may also be masking much of the detailed information needed to located and handle branch points. The SNR is now increased to 60.

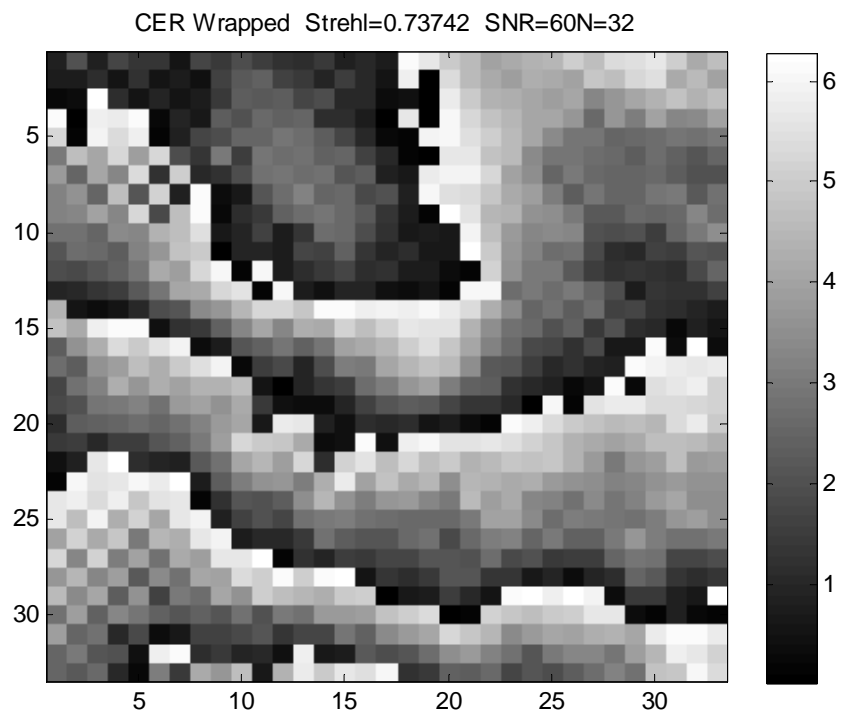


Figure 33. NVWER. $S=0.737$, $\text{SNR}=60$, and $N=32$

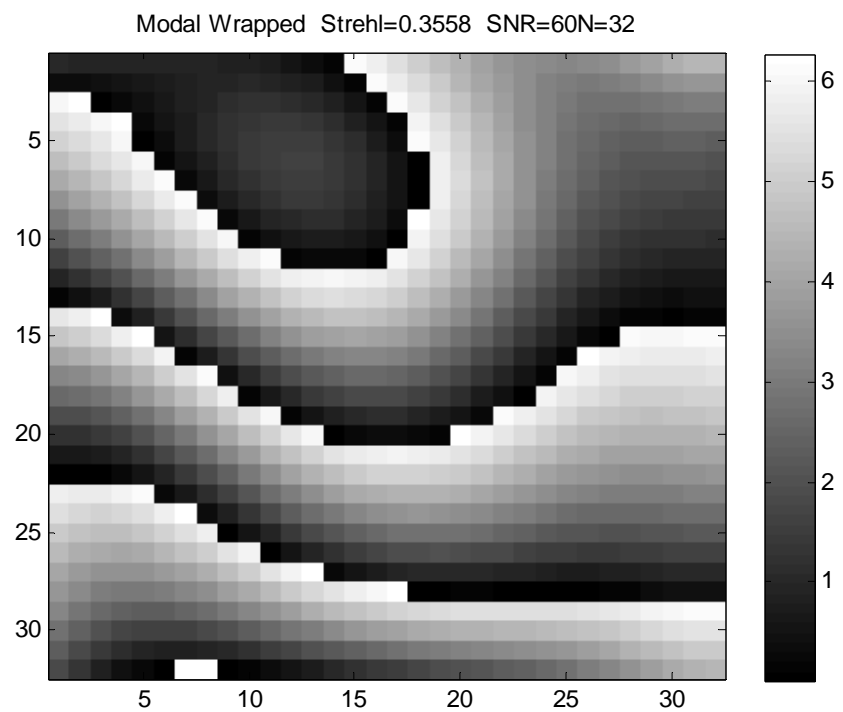


Figure 34. Modal Reconstruction. $S=0.356$, $\text{SNR}=60$, and $N=32$

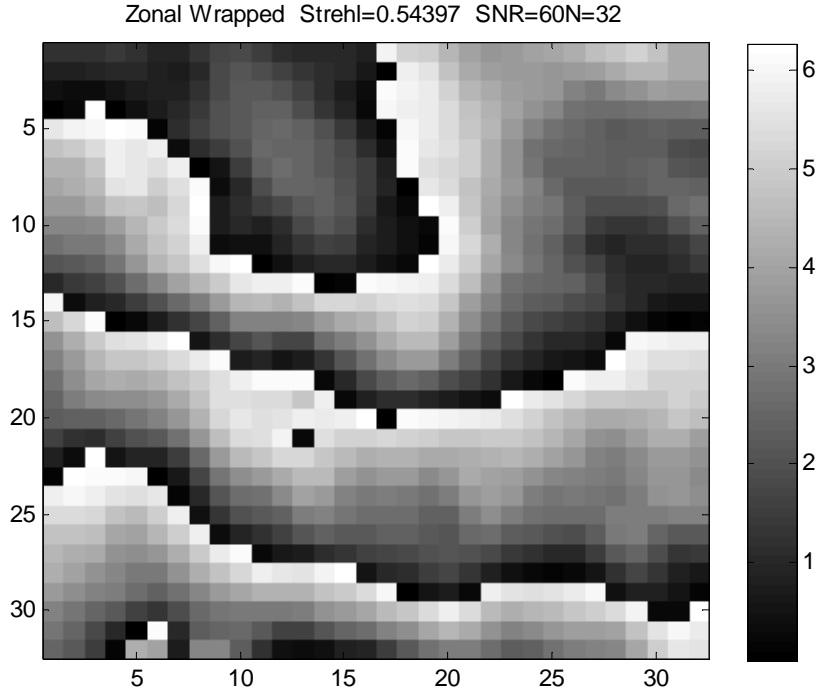


Figure 35. Zonal Reconstruction. $S=0.544$, $SNR=60$, and $N=32$

When the SNR is increased to 60, representing a relatively sensitive SHWFS but still subjected to noise, all of the reconstructors increase in Strehl ratio. The NVWCER performed the best with a Strehl of 0.737, with the zonal performing second with a Strehl of 0.544, and modal performing the worst with a Strehl of 0.356. With the NVWCER receiving less noisy data, it is able to perform as expected in the presence of deep turbulence. The last section is an SNR of 200, which represents a nearly ideal SHWFS.

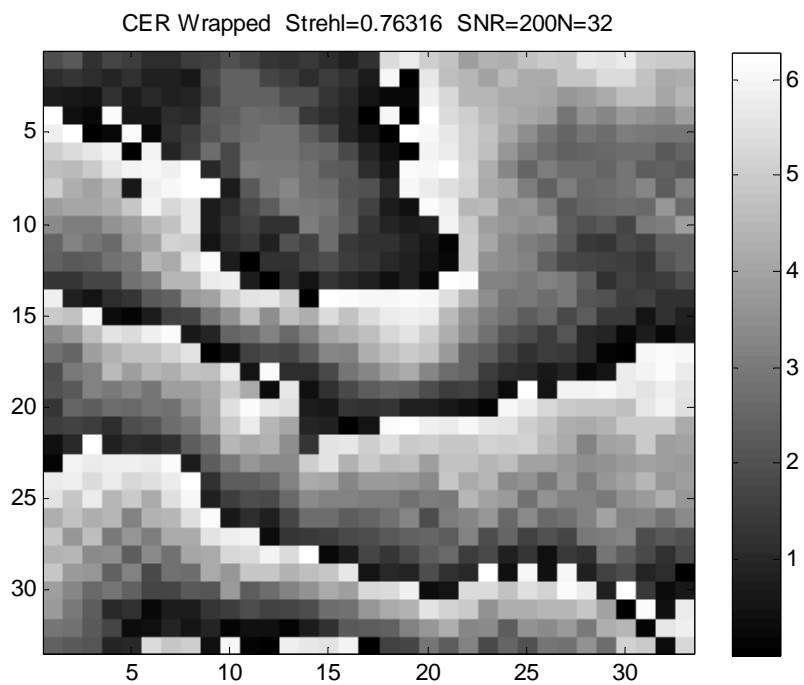


Figure 36. NVWCER. $S=0.763$, $SNR=200$, and $N=32$

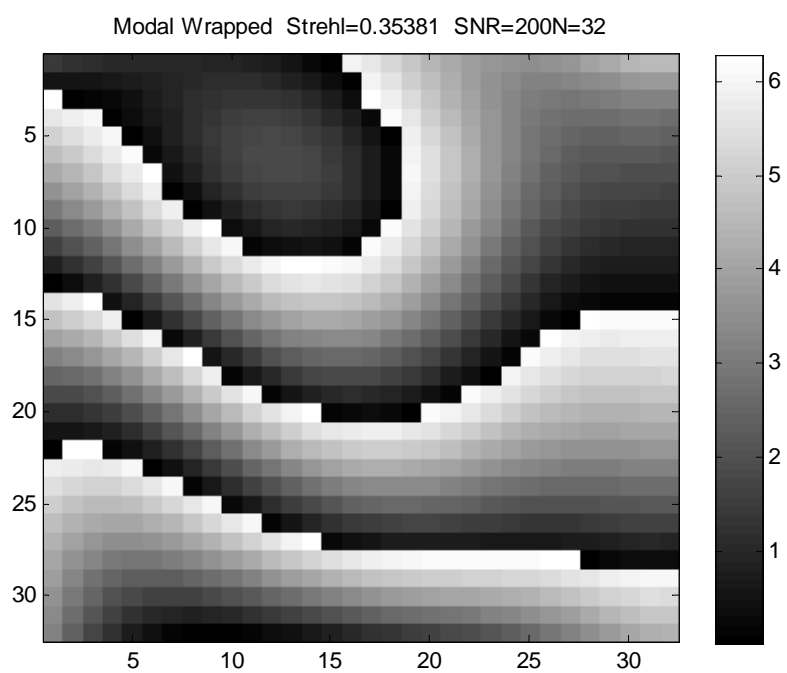


Figure 37. Modal Reconstruction. $S=0.354$, $SNR=200$, and $N=32$

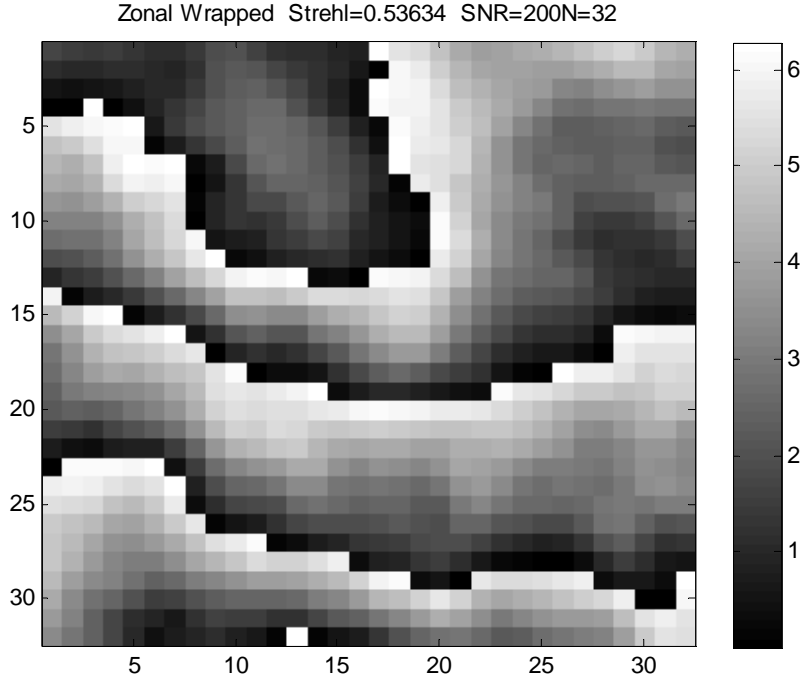


Figure 38. Zonal Reconstruction. $S=0.536$, $\text{SNR}=200$, and $N=32$

A SNR of 200 yielded minimal differences from a value of 60, with only a slight increase in the Strehl for the NVWCER. The reconstructors already performed decently with an SNR of 60, and the noise was not the driving factor at that point. It was just the ability of the sensors to handle the SHWFS data in the presence of branch points. Since the NVWCER is designed to handle branch points, it makes sense that it would be the one to gain the most from a higher SNR, then the other reconstructors.

The variations in both SNR and number of sensors have yielded interesting results, and from them, the data presented to compare all of the reconstructors will use an SNR of 60 and an array of sensors with length 32 on a side.

C. EVALUATION OF RECONSTRUCTORS

The reconstructors are evaluated for each Rytov variance, with a constant SNR and number of sensors. The zonal and modal reconstructors are designed to work in environments with little to no turbulence, while the NVWCER is designed specifically to handle branch points, as a result of high turbulence. The 0.109 Rytov variance represents

an area of low turbulence, so all reconstructors should perform similarly, while the variances of 0.349 and 0.567 are areas of higher turbulence, where the NVWCER should outperform the others.

1. Rytov Variance of 0.109

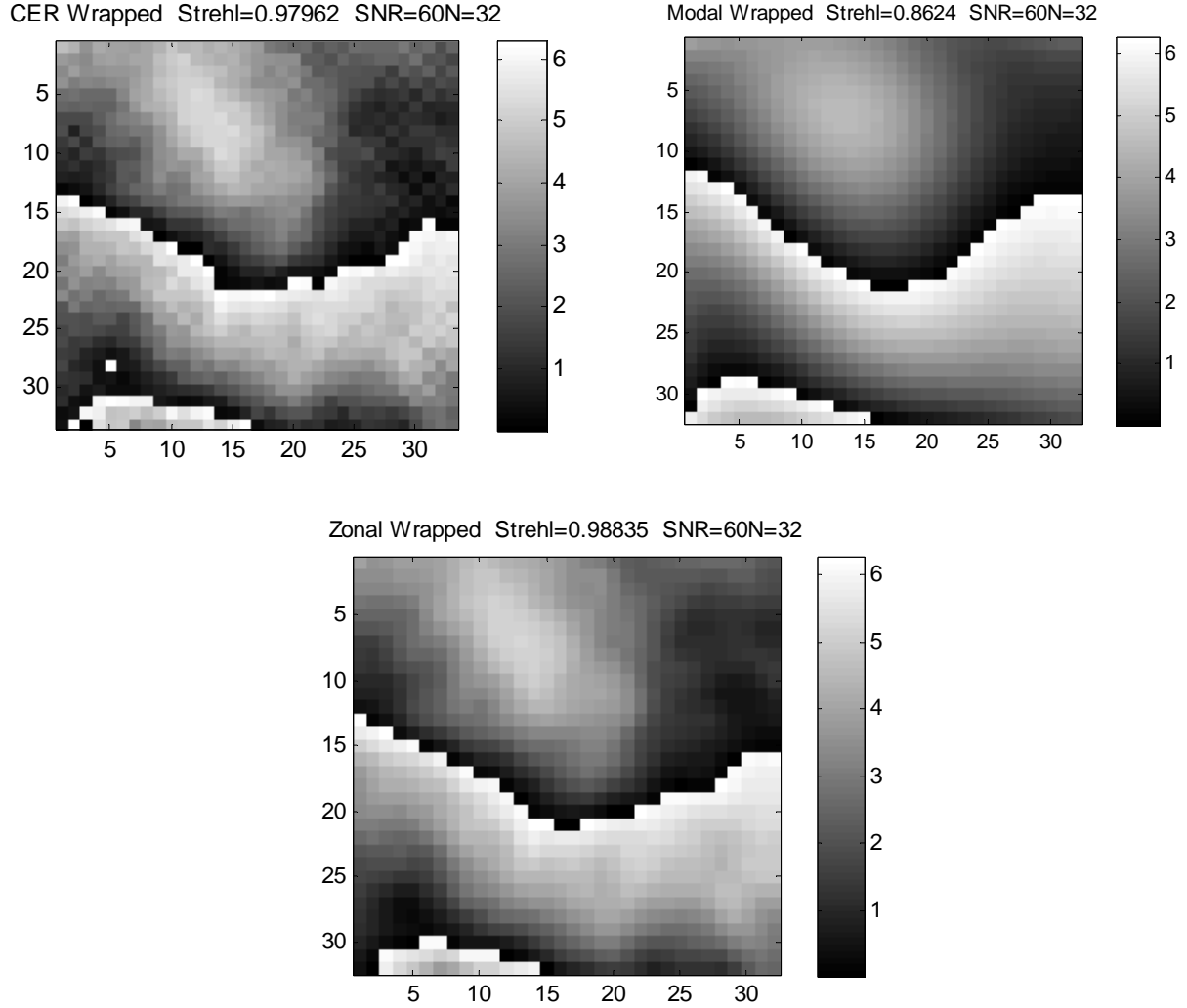


Figure 39. Reconstructor Comparison for Rytov of 0.109. SNR=60 and N=32

With a low about of turbulence, the reconstructors performed as predicted, with the zonal reconstructors and NVWCER performing nearly the same with an almost a perfect Strehl ratio; 0.977 and 0.988, respectively. The modal reconstructor is still a

victim of its inherent smoothing leading to a lower, 0.862, Strehl ratio, with a limited number of modes used.

2. Rytov Variance of 0.349

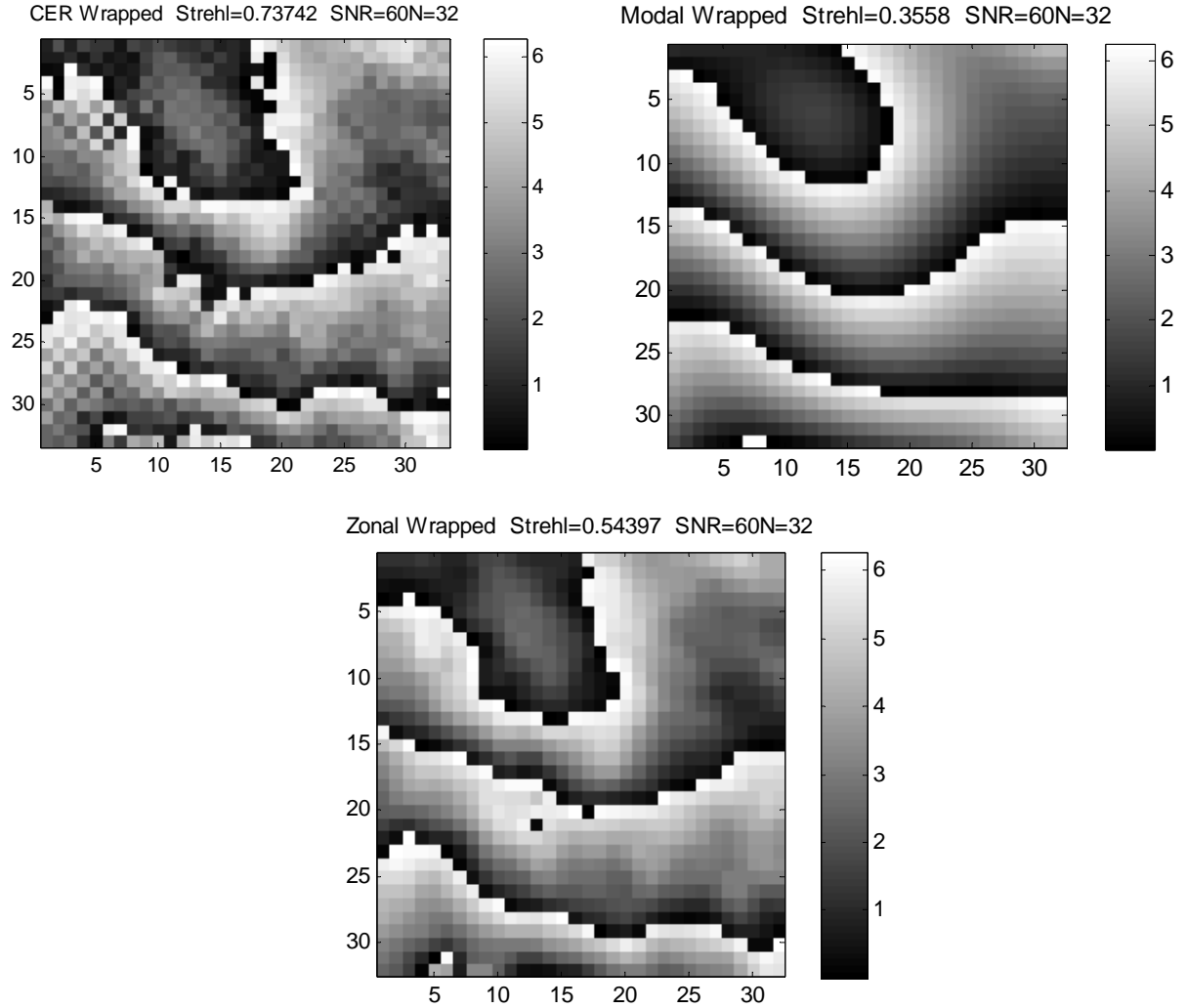


Figure 40. Reconstructor Comparison for Rytov of 0.349. SNR=60 and N=32

When the turbulence is increased the effects of each reconstructor are more prominent. The modal performs significantly less, with a Strehl of 0.356. The NVWCER is outperforms the zonal reconstructor, with Strehl ratios of 0.737 and 0.544, respectively, due the greater presence of branch points and branch cuts. Since the zonal reconstructor

does not account for branch points, the more of them present, the greater the error of the reconstructor.

3. Rytov Variance of 0.567

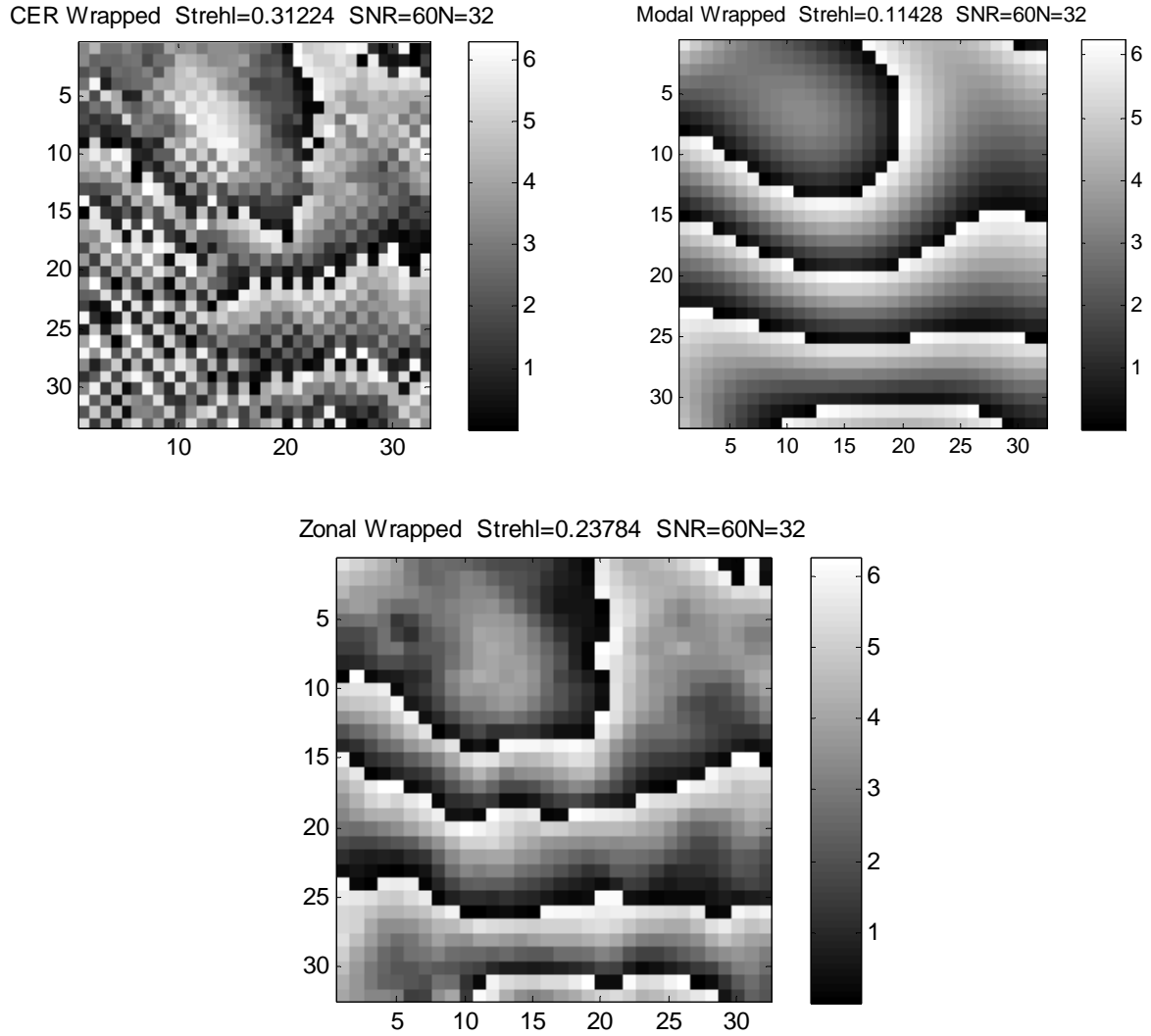


Figure 41. Reconstructor Comparison for Rytov of 0.567. SNR=60 and N=32

When the turbulence was increased again, to a Rytov variance of 0.567, all three reconstructors drop significantly in the Strehl ratio. The modal reconstructor performed the worst again, with a Strehl of 0.114, while the NVWCER outperformed the zonal reconstructor with Strehl's of 0.312 and 0.238 accordingly. The ability of any

reconstructor greatly falls off when the turbulence increases beyond a certain threshold, and it is no longer an issue of optical correction, rather an inappropriate operational environment.

D. FINAL RESULTS

Table 3 is the final results of all simulations ran for complete comparison and evaluation of all three reconstructors. All permutations of possible SNR (10,60 and 200), Rytov variances (0.109, 0.349, and 0.567), and number of sensors per side (16, 32, and 64) were ran and evaluated.

	Strehl Ratio									
	Sensors	16			32			64		
	SNR	10	60	200	10	60	200	10	60	200
Zonal	0.109	0.8461	0.9699	0.9649	0.8645	0.9884	0.9897	0.8350	0.9887	0.9968
	0.349	0.1812	0.4055	0.4594	0.2304	.5292	0.5403	0.2205	0.7015	0.6178
	0.567	0.1201	0.1405	0.1330	0.1515	0.2035	0.2052	0.0920	0.2295	0.2312
Modal	0.109	0.7800	0.8597	0.8574	0.7796	0.8624	0.8626	0.7863	0.8560	0.8648
	0.349	0.1273	0.2886	0.3277	0.1561	0.3445	0.3513	0.1402	0.4370	0.3929
	0.567	0.0478	0.0809	0.0714	0.0943	0.1057	0.1082	0.0437	0.1237	0.1208
NVWCER	0.109	0.6016	0.9538	0.9584	0.6331	0.9796	0.9873	0.2539	0.9805	0.9915
	0.349	0.0187	0.5557	0.5665	0.2063	0.7831	0.7724	0.1328	0.8737	0.8907
	0.567	0.0877	0.1609	0.1882	0.1093	0.3696	0.4017	0.2093	0.6266	0.3942
	Rytov Variance									

Table 3. Results for all Reconstruction Simulations

VI. CONCLUSION

This research was aimed at computationally evaluating the performance of two well-known wave front reconstruction methods, zonal and modal, against a modern reconstruction method, a Noise-Variance Weighted Complex Exponential Reconstructor, created by Dr. David Fried, in the presence of deep turbulence. The NVWCER is designed to reconstruct using a complex exponential, so it can utilize both the hidden and scalar phase, allowing it better handle branch points. The reconstructors were evaluated using simulated SHFWS data using various combinations of Rytov variances, SNR, and number of sensors.

After testing an evaluation the NVWCER has shown its ability to outperform the modal and zonal reconstruction methods in the presence of deep turbulence. At a Rytov variance of 0.109, representing weak turbulence, all three reconstructors performed well. When the turbulence level was increased the NVWCER continuously performed better than both other reconstructors. The zonal reconstruction method typically outperformed the modal as well, due to its inherent smoothing. There are some outlying effects such as when the number of sensors was increased greatly; it sometimes had a negative effect on the NVWCER. The reasoning for this can only be postulated presently, as the sample points being smaller than the branch point themselves. The effects of “egg-crating” also were a source of error for the NVWCER, which was due the use of two lattices and bringing them into phase with each other, in order to operate with the SHFWS geometry.

This research shows promise for the use of a NVECER algorithm in environments with deep turbulence, like a maritime environment. Two areas of concern and interest for future work involve the computational efficiency of the NVWCER as well as a different sensor pairing. Currently the NVWCER is an iterative process, using many for loops and minimal matrix operations, and it would be useful to have some form of a parallel process designed. The most important issue with the NVWCER, and its use with a SHFWS, is the differences in geometries. The NVWCER is designed for a Hudgin geometry, and later altered to work with Fried or Southwell geometry, however that introduces the lattice phase issues, which in turn induce error. Either a shearing interferometer needs to be

paired with the NVWCER, or the algorithm needs to be adapted to work directly with Fried or Southwell geometry.

APPENDIX.

A. INITIALIZATION AND SIMULATION CODE

```
close all;
clear all;
clc;

%% Phase Loading
% Choose one for random phase generation or 2 for MyPhs data
vv = 3;

if vv==3
    load('amp_n_phase_0.567_1.mat');
    phase2pi=phase2pi+pi;
    N=9;
    phil=phase2pi;
end
SNR0=60;
NN=2^6;
NoiseFlag=1; %1 adds noise. 0 does NOT add noise

%[Fx,Fy,Magnitudes,SNR] =
slope_WgtAvg(phase2pi,ones(size(phase2pi)),NN,SNR0,NoiseFlag);
[Fx,Fy,Magnitudes,SNR] =
slope_WgtAvg(phase2pi,amplitude2pi,NN,SNR0,NoiseFlag);

VLQx=1./Magnitudes;
VLQy=VLQx;
h=1;
dx=size(phase2pi,1)/NN;
[E,phase_CER,VLQ]=NVWCER_SHWFS_V2(Fx,Fy,VLQx,VLQy);
phase_zonal=zonal_2(Fx,Fy,h);
[PhaseModal,Terms]=ZernikeModal(Fx*dx,Fy*dx);

%% Strehl Ratio Calculations method 1;
ims=size(phase2pi,1);
dd=ims/NN; %Spacing for each subsample distance

undersamp_sw=amplitude2pi(dd/2:dd:ims-dd/2,dd/2:dd:ims-
dd/2).*exp(i*phase2pi(dd/2:dd:ims-dd/2,dd/2:dd:ims-dd/2));
undersamp_fr=amplitude2pi([1:dd:ims ims],[1:dd:ims
ims]).*exp(i*phase2pi([1:dd:ims ims],[1:dd:ims ims]));

reff_sw=(sum(sum(abs(undersamp_sw))))^2;
reff_fr=(sum(sum(abs(undersamp_fr))))^2;
```

```

comp_zonal=exp(-i*phase_zonal);
comp_modal=exp(-i*PhaseModal);
comp_CER_unwrap=exp(-i*phase_CER);
comp_CER_wrap=conj(E)./abs(E);

strehl_zonal=(abs(sum(sum(undersamp_sw.*comp_zonal))))^2/reff_sw;
strehl_modal=(abs(sum(sum(undersamp_sw.*comp_modal))))^2/reff_sw;
strehl_CER_unwrap=(abs(sum(sum(undersamp_fr.*comp_CER_unwrap))))^2/reff_fr;
strehl_CER_wrap=(abs(sum(sum(undersamp_fr.*comp_CER_wrap))))^2/reff_fr;
%% Plots
eval(['cd 567\' num2str(SNR0) \'\' num2str(NN)])
figure;imagesc(phase_CER);title(['CER Unwrapped ' 'Strehl='
num2str(strehl_CER_unwrap) ' SNR=' num2str(SNR0) 'N='
num2str(NN)]);colorbar; colormap gray;
saveas(gcf,'CER_Unwrapped.fig','fig');
figure;imagesc(phase_zonal);title(['Zonal Unwrapped ' 'Strehl='
num2str(strehl_zonal) ' SNR=' num2str(SNR0) 'N='
num2str(NN)]);colorbar; colormap gray;
saveas(gcf,'Zonal_Unwrapped.fig','fig');
figure;imagesc(PhaseModal);title(['Modal Unwrapped ' 'Strehl='
num2str(strehl_modal) ' SNR=' num2str(SNR0) 'N='
num2str(NN)]);colorbar; colormap gray;
saveas(gcf,'Modal_Unwrapped.fig','fig');
figure;imagesc(mod(phase_CER,2*pi));title(['CER Wrapped ' 'Strehl='
num2str(strehl_CER_unwrap) ' SNR=' num2str(SNR0) 'N='
num2str(NN)]);colorbar; colormap gray;
saveas(gcf,'CER_Wrapped.fig','fig');
figure;imagesc(mod(phase_zonal,2*pi));title(['Zonal Wrapped '
'Strehl=' num2str(strehl_zonal) ' SNR=' num2str(SNR0) 'N='
num2str(NN)]);colorbar; colormap gray;
saveas(gcf,'Zonal_Wrapped.fig','fig');
figure;imagesc(mod(PhaseModal,2*pi));title(['Modal Wrapped ' 'Strehl='
num2str(strehl_modal) ' SNR=' num2str(SNR0) 'N='
num2str(NN)]);colorbar; colormap gray;
saveas(gcf,'Modal_Wrapped.fig','fig');
cd ../../../../;

```

B. SHACK-HARTMANN WAVEFRONT SENSOR SIMULATION CODE

```

function [Fx,Fy,Magnitudes,SNR] =
slope_WgtAvg(phase2pi,amplitude2pi,NN,SNR0,flag)

%Create Complex Matrix from amplitude and phse
Phi_Complex=amplitude2pi.*exp(1i*phase2pi);
%Generate complex differentials
PhiDx=Phi_Complex(:,2:end).*conj(Phi_Complex(:,1:end-1));
PhiDy=Phi_Complex(2:end,:).*conj(Phi_Complex(1:end-1,:));

%Magnitude of differentials for weighting
MagX=abs(PhiDx);
MagY=abs(PhiDy);

```

```

%Phase of differences
thetaX=angle(PhiDx);
thetaY=angle(PhiDy);

Intensity=abs(Phi_Complex).^2;
Intensity=[Intensity(:,1) Intensity Intensity(:,end)];
Intensity=[Intensity(1,:); Intensity; Intensity(end,:)];

%repeat edge values
MagX=[MagX(:,1) MagX MagX(:,end)];
MagY=[MagY(1,:); MagY; MagY(end,:)];
thetaX=[thetaX(:,1) thetaX thetaX(:,end)];
thetaY=[thetaY(1,:); thetaY; thetaY(end,:)];

%Calculate differential size
[y,x]=size(phase2pi);
dy=round(y/NN);
dx=round(x/NN);

%Slope outputs
Fx=zeros(NN);
Fy=zeros(NN);
Magnitudes=zeros(NN);

%Weighting Factors
%Since the column (row) for x (y) differentials has overlapping of
sample
%area for each lenslet and is ideally located in the middle of each
phase
%point (fried geometry) a weight of one half is applied to the over
sampled
%values on the edges
Wx=ones(dy,dx+1);
Wx(:,1)=0.5;
Wx(:,end)=0.5;

Wy=ones(dy+1,dx);
Wy(1,:)=0.5;
Wy(end,:)=0.5;

Wmag=ones(dy+1,dx+1);
Wmag(:,1)=0.5;
Wmag(:,end)=0.5;
Wmag(1,:)=Wmag(1,:).*0.5;
Wmag(end,:)=Wmag(end,:).*0.5;

thetaFx=zeros(dy,dx+1);
thetaFy=zeros(dy+1,dx);

magFx=zeros(dy,dx+1);
magFy=zeros(dy+1,dx);

```

```

for u=1:NN
    for v=1:NN
        thetaFx=thetaX((v-1)*dy+1:v*dy,(u-1)*dx+1:u*dx+1);
        thetaFy=thetaY((v-1)*dy+1:v*dy+1,(u-1)*dx+1:u*dx);
        magFx=MagX((v-1)*dy+1:v*dy,(u-1)*dx+1:u*dx+1).*Wx;
        magFy=MagY((v-1)*dy+1:v*dy+1,(u-1)*dx+1:u*dx).*Wy;
        Fx(v,u)=sum(sum(thetaFx.*magFx))/sum(sum(magFx)).*dx; %tiltx
        Fy(v,u)=sum(sum(thetaFy.*magFy))/sum(sum(magFy)).*dy; %tilty
        Magnitudes(v,u)=sum(sum(Intensity((v-1)*dy+1:v*dy+1,(u-
1)*dx+1:u*dx+1).*Wmag));
    end
end

SNR=SNR0*Magnitudes/mean(Magnitudes(:));

delX=randn(NN,NN).*pi./SNR;
delY=randn(NN,NN).*pi./SNR;

if flag==1
    Fx=Fx+delX;
    Fy=Fy+delY;
end

end

```

C. ZONAL RECONSTRUCTION

```

function [phases] = zonal_2(Fx,Fy,h)

[a,b]=size(Fx);
[c,d]=size(Fy);

if c~=a || d~=a
    error('Size of Fx differs from Fy');
end
if a~=b
    error('Fx is not square');
end
if c~=d
    error('Fy is not square');
end

N=a;

S=[reshape(Fx',1,N^2) reshape(Fy',1,N^2)]';

A=formA(N);
D=formD(N);

phases=pinv(A)*D*S;

% [U,A2,V]=svd(A,0);

```

```

% A2=pinv(A2);
% phases=V*A2*U'*D*S;

phases=reshape(phases,N,N)'./h;
end

% Derived from (Dai 2008)
function A=formA(N)
A=zeros(2*N*(N-1),N^2);
for i=1:N
    for j=1:(N-1)
        A((i-1)*(N-1)+j,(i-1)*N+j)=-1;
        A((i-1)*(N-1)+j,(i-1)*N+j+1)=1;
        A((N+i-1)*(N-1)+j,i+(j-1)*N)=-1;
        A((N+i-1)*(N-1)+j,i+j*N)=1;
    end
end
end

function D=formD(N)
D=zeros(2*N*(N-1),2*N^2);
for i=1:N
    for j=1:(N-1)
        D((i-1)*(N-1)+j,(i-1)*N+j)=0.5;
        D((i-1)*(N-1)+j,(i-1)*N+j+1)=0.5;
        D((N+i-1)*(N-1)+j,N*(N+j-1)+i)=0.5;
        D((N+i-1)*(N-1)+j,N*(N+j)+i)=0.5;
    end
end
end
end

```

D. MODAL RECONSTRUCTION

```

function [PhasesModal,terms]=ZernikeModal(Fx,Fy)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%% Adapted from Matthew Allen Modal Code (c)
2007%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

NN=size(Fx,1);

[x , y]=meshgrid(-1+(2/(2*NN)):2/(NN):1);
x=reshape(x,[],1);
y=reshape(y,[],1);

S=[reshape(Fx,[],1); reshape(Fy,[],1)];

```

```

x_2 = x.^2;
x_3 = x.^3;
x_4 = x.^4;
x_5 = x.^5;
y_2 = y.^2;
y_3 = y.^3;
y_4 = y.^4;
y_5 = y.^5;

dZdx=zeros(size(x,1),21);
dZdy=zeros(size(y,1),21);
%Partial Derivatives Evaluation dX

dZdx(:,1)=1;
dZdx(:,2)=0;
dZdx(:,3)=4.*x;
dZdx(:,4)=2.*x;
dZdx(:,5)=2.*y;
dZdx(:,6)=-2+9.*x_2+3.*y_2;
dZdx(:,7)=6.*x.*y;
dZdx(:,8)=-12.*x+24.*x.*(x_2+y_2);
dZdx(:,9)=3.*x_2-3.*y_2;
dZdx(:,10)=6.*x.*y;
dZdx(:,11)=-6.*x+8.*x.*(x_2+y_2)+8.*x_3-8.*x.*y_2;
dZdx(:,12)=-6.*y+8.*y.*(x_2+y_2)+16.*x_2.*y;
dZdx(:,13)=3-36.*x_2-12.*y_2+10.*(x_2+y_2).^2+40.*x_2.*(x_2+y_2);
dZdx(:,14)=-24.*x.*y+40.*x.*y.*(x_2+y_2);
dZdx(:,15)=24.*x-120.*x.*(x_2+y_2)+120.*x.*(x_2+y_2).^2;
dZdx(:,16)=4.*x_3-12.*x.*y_2;
dZdx(:,17)=12.*x_2.*y-4.*y_3;
dZdx(:,18)=-12.*x_2+12.*y_2+15.*x_2.*(x_2+y_2)+10.*x_4-
15.*y_2.*(x_2+y_2)-30.*x_2.*y_2;
dZdx(:,19)=-24.*x.*y+30.*x.*y.*(x_2+y_2)+30.*x_3.*y-10.*x.*y_3;
dZdx(:,20)=12.*x-40.*x.*(x_2+y_2)-
40.*x_3+40.*x.*y_2+30.*x.*(x_2+y_2).^2+60.*x_3.*(x_2+y_2)-
60.*x.*y_2.*(x_2+y_2);
dZdx(:,21)=12.*y-40.*y.*(x_2+y_2)-
80.*x_2.*y+30.*y.*(x_2+y_2).^2+120.*x_2.*y.*(x_2+y_2);
% dZdx(sqrt(x_2+y_2)>1,:)=0;
%Partial Derivatives Evaluation dY
dZdy(:,1)=0;
dZdy(:,2)=1;
dZdy(:,3)=4.*y;
dZdy(:,4)=-2.*y;
dZdy(:,5)=2.*x;
dZdy(:,6)=6.*x.*y;
dZdy(:,7)=-2+3.*x_2+9.*y_2;
dZdy(:,8)=-12.*y+24.*y.*(x_2+y_2);
dZdy(:,9)=-6.*x.*y;
dZdy(:,10)=3.*x_2-3.*y_2;
dZdy(:,11)=6.*y+8.*x_2.*y-8.*y.*(x_2+y_2)-8.*y_3;
dZdy(:,12)=-6.*x+8.*x.*(x_2+y_2)+16.*x.*y_2;
dZdy(:,13)=-24.*x.*y+40.*x.*y.*(x_2+y_2);
dZdy(:,14)=3-12.*x_2-36.*y_2+10.*(x_2+y_2).^2+40.*y_2.*(x_2+y_2);

```

```

dZdy(:,15)=24.*y-120.*y.*(x_2+y_2)+120.*y.*(x_2+y_2).^2;
dZdy(:,16)=-12.*x_2.*y+4.*y_3;
dZdy(:,17)=4.*x_3-12.*x.*y_2;
dZdy(:,18)=24.*x.*y+10.*x_3.*y-30.*x.*y.*(x_2+y_2)-30.*x.*y_3;
dZdy(:,19)=-12.*x_2+12.*y_2+15.*x_2.*(x_2+y_2)+30.*x_2.*y_2-
15.*y_2.*(x_2+y_2)-10.*y_4;
dZdy(:,20)=-12.*y-
40.*x_2.*y+40.*y.*(x_2+y_2)+40.*y_3+60.*x_2.*y.*(x_2+y_2)-
30.*y.*(x_2+y_2).^2-60.*y_3.*(x_2+y_2);
dZdy(:,21)=12.*x-40.*x.*(x_2+y_2)-
80.*x.*y_2+30.*x.*(x_2+y_2).^2+120.*x.*y_2.*(x_2+y_2);
% dZdy(sqrt(x_2+y_2)>1,:)=0;

dZ=[dZdx;dZdy];

terms=pinv(dZ)*S;
a=terms;

[x, y]=meshgrid((-1+2/NN/2):2/NN:(1-2/NN/2));

% [x y]=meshgrid(-1:2/NN:1);

x_2 = x.^2;
x_3 = x.^3;
x_4 = x.^4;
x_5 = x.^5;
y_2 = y.^2;
y_3 = y.^3;
y_4 = y.^4;
y_5 = y.^5;

PhasesModal=a(1)*x+...
a(2)*y+...
a(3)*(-1+2.*(x_2+y_2))+...
a(4)*(x_2-y_2)+...
a(5)*(2.*x.*y)+...
a(6)*(-2.*x+3.*x.*(x_2+y_2))+...
a(7)*(-2.*y+3.*y.*(x_2+y_2))+...
a(8)*(1-6.*(x_2+y_2)+6.*(x_2+y_2).^2)+...
a(9)*(x_3-3.*x.*y.^2)+...
a(10)*(3.*x_2.*y-y_3)+...
a(11)*(-3.*x_2+3.*y_2+4.*x_2.*(x_2+y_2)-4.*y_2.*(x_2+y_2))+...
a(12)*(-6.*x.*y+8.*x.*y.*(x_2+y_2))+...
a(13)*(3.*x-12.*x.*(x_2+y_2)+10.*x.*(x_2+y_2).^2)+...
a(14)*(3.*y-12.*y.*(x_2+y_2)+10.*y.*(x_2+y_2).^2)+...
a(15)*(-1+12.*(x_2+y_2)-30.*(x_2+y_2).^2+20.*(x_2+y_2).^3)+...
a(16)*(x_4-6.*x_2.*y_2+y_4)+...
a(17)*(4.*x_3.*y-4.*x.*y_3)+...
a(18)*(-4.*x_3+12.*x.*y_2+5.*x_3.*(x_2+y_2)-15.*x.*y_2.*(x_2+y_2))+...
a(19)*(-12.*x_2.*y+4.*y_3+15.*x_2.*y.*(x_2+y_2)-5.*y_3.*(x_2+y_2))+...
a(20)*(6.*x_2-6.*y_2-
20.*x_2.*(x_2+y_2)+20.*y_2.*(x_2+y_2)+15.*x_2.*(x_2+y_2).^2-
15.*y_2.*(x_2+y_2).^2)+...

```



```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%% Assign Hartman data to first lattice and carry out reconstruction
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

NN=ceil(log2(a));                                %Creates proper size 2^N lattice
L=2^NN;                                           %Number of unit squares to be used
dpu=zeros(L+1,L);
dpv=zeros(L,L+1);
sigsqu=inf*ones(L+1,L);
sigsqv=inf*ones(L,L+1);

%assigning values from Shack Hartmann Sensor to first lattice for
%exponential reconstructor.
for x=1:a
    for y=1:a
        if x+y==2*round((x+y)/2)
            u=(x-y+L)/2+1;
            v=(x+y)/2;
            dpv(v,u)=dpx(y,x)+dpy(y,x);
            sigsqv(v,u)=sigsqx(y,x)+sigsqy(y,x);
        else
            u=(x-y+L+1)/2;
            v=(x+y+1)/2;
            dpu(v,u)=dpx(y,x)-dpy(y,x);
            sigsqv(v,u)=sigsqx(y,x)+sigsqy(y,x);
        end
    end
end

%Hudgin lattice generated for phase 1. Passed into NWCER with
differential
%phasors calculated in the equation (exp(i*dp#))
[Ep,sigsqp]=NWCER4SID(exp(i*dpu),exp(i*dpv),sigsqu,sigsqv);

%Create raw lattices so that only values which correpond to the
original
%lattice are taken from the hudgin geometry generation and calculation
E1=NaN*ones(a+1,a+1);
sigsq1=NaN*ones(a+1,a+1);

%Go through point-by-point on large (hudgin/phase 1 lattice) and assign
%only values which correspond to values overlapping with the original
%lattice. If they are outside the appearture they maintain the value of
0
%phase and Inf variance (noise)

for u=1:L+1

```

```

    for v=1:L+1
        x=u+v-1-L/2;
        y=-u+v+1+L/2;
        if ((x>=1 & x<=a+1) & (y>=1 & y<=a+1)) & isfinite(sigsqp(u,v))
            E1(y,x)=Ep(v,u);
            sigsql(y,x)=sigsqp(v,u);
        end
    end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Assign Hartman data to second lattace and carry out reconstruction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

dpu=zeros(L+1,L);
dpv=zeros(L,L+1);
sigsqu=inf*ones(L+1,L);
sigsqv=inf*ones(L,L+1);

for x=1:a
    for y=1:a
        if x+y==2*round((x+y)/2)
            u=(x-y+L)/2;
            v=(x+y)/2;
            dpu(v,u)=dpx(y,x)-dpy(y,x);
            sigsqu(v,u)=sigsqx(y,x)+sigsqy(y,x);
        else
            u=(x-y+L+1)/2;
            v=(x+y-1)/2;
            dpv(v,u)=dpx(y,x)+dpy(y,x);
            sigsqv(v,u)=sigsqx(y,x)+sigsqy(y,x);
        end
    end
end

[Epp,sigsqpp]=NVWCER4SID(exp(i*dpu),exp(i*dpv),sigsqu,sigsqv);

E2=NaN*ones(a+1,a+1);
sigsq2=NaN*ones(a+1,a+1);

for u=1:L+1
    for v=1:L+1
        x=u+v-L/2;
        y=-u+v+1+L/2;
        if ((x>=1 & x<=a+1) & (y>=1 & y<=a+1)) & isfinite(sigsqp(u,v))
            E2(y,x)=Epp(v,u);

```

```

        sigsq2(y,x)=sigsqpp(v,u);
    end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%
%%% MERGE MATRICES BACK TOGETHER INTO ORGINAL
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

E2t=NaN*ones(a+1);
sigsq2t=NaN*ones(a+1);
for x=1:a+1
    for y=1:a+1
        if x+y~=2*round((x+y)/2)
            S=0; T=0;
            if x-1>=1
                S=E1(y,x-1)/sigsq1(y,x-1);
                T=1/sigsq1(y,x-1);
            end
            if x+1<=a+1
                S=S+E1(y,x+1)/sigsq1(y,x+1);
                T=T+1/sigsq1(y,x+1);
            end
            if y-1>=1
                S=S+E1(y-1,x)/sigsq1(y-1,x);
                T=T+1/sigsq1(y-1,x);
            end
            if y+1<=a+1
                S=S+E1(y+1,x)/sigsq1(y+1,x);
                T=T+1/sigsq1(y+1,x);
            end
            T=4/T;
            S=S/(4/T);
            E2t(y,x)=S;
            sigsq2t(y,x)=T;
        end
    end
end

[x,y]=meshgrid(1:a+1);
ii=find(x+y~=2*round((x+y)/2));
S=sum(E2t(ii).*conj(E2(ii)).*mask(ii)./(sigsq2t(ii)+sigsq2(ii)));
phi2l=angle(S);
E=E1;

```

```

E(ii)=E2(ii)*exp(i*phi21);
VLQ=sigsq1;
VLQ(ii)=sigsq2(ii);
jj=find(~isfinite(VLQ));
E(jj)=NaN;

E=E./abs(E);
if nargin==6
    E=amp.*E;
end

phi=BranchPointPhase(E);

end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%
%%
% Function, called BranchPointPhase, used to generate a phase function,
% PHI,
% that corresponds (modulo 2*pi) to the phase of the input complex
% phasor,U,
% and that is---as far as possible---smooth, i.e., which minimizes
% difference
% between values of phi between adjacent points in the array of values
% of
% phi. Ideally, the magnitude of the difference of phase values between
% adjacent points should be less than pi everywhere except at branch-
% cuts.
%
% This function is formed as the combination of the two functions
% SmoothPhase
% and Reconstructor that are listed in TN-100. The contribution that
% comes
% from (or rather that was) the function called Reconstructor appears
% here as
% a sub function, clearly identified as being Reconstructor, in the
% final
% part of this listing. All the rest of the listing comes, essentially
% intact, from SmoothPhase.
%
% The input function, U, defines the extent of the aperture by its
% value
% being equal to NaN for those array points corresponding to positions
% outside the aperture. The array on which U is defined is to be
% square, with
% U corresponding to a clear circular aperture just smaller than the
% size of
% the array. For points on the array that are within the aperture the
% values
% of U are complex numbers (representing phasors of the optical field)
% while
% for points on the array that are outside the aperture the values of U
% are

```

```

% NaN.
%
% The calculation of PHI is accomplished by first evaluating the
difference
% of the phase (modulo  $2\pi$ ) between adjacent values of U, using those
phase
% differences, dPx and dPy, to calculate the curl for each elemental
square
% of the array, and using the values of the curl to determine the
location
% and sign of the branch points. Using this branch point information
the
% complex phaseor, Uh, associated with the branch points is calculated
and
% from this the hidden phase, phi, is then calculated. The ratio U over
Uh,
% denoted by u, is calculated. The phase differences dpx and dpy
associated
% with u are evaluated. Then the associated phase is generated by first
% adding/subtracting dpy values along the central line of the array to
give
% phase values for the all the points above/below the center point of
the
% central line, and then starting from each point on that central line
by
% adding/subtracting dpx values the phase value for each point to the
% right/left of the central point line the phase value for each point
in the
% array is calculated. To these phase values, which may be associated
with
% the non hidden phase values, the hidden phase values, phi, are added
to
% yield the final output phase values, PHI.
%
% For those cases in which there the number of positive sign branch
points is
% not equal to the number of negative sign branch points (presumably
because
% the "missing" branch point is outside the aperture and so is not
observed)
% an additional branch point of the appropriate sign is provided. This
branch
% point is placed at a position just outside the aperture and at a
location
% chosen on the basis of a "potential field," V, formed by the positive
and
% negative sign branch points. The position is chosen to correspond to
the
% maximum of the potential field---the maximum over positions just
outside
% the aperture. This added branch point is introduced to "guide" the
branch
% cut that goes out of the aperture.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

```

```

%
%      INPUTS
%   U      (N,M) = Complex phasor array. (The values of U are equal to
NaN
%           for those array points whose position is outside
the
%           aperture.)
%
%      OUTPUTS
%   PHI     (N,M) = Maximally smooth phase corresponding (modulo 2*pi)
to the
%           phase of U. (Values of PHI are equal to NaN for
points
%           outside the aperture.)
%   BPcount (1,1) = Number of branch points in U (located within the
%           aperture). respectively.
%   phi     (N,M) = Hidden phase corresponding to U. (Values of phi are
equal
%           to NaN for points outside the aperture.)
%   BPes    (a,3) = The second column of values gives the x-axis (i.e.,
column
%           number) position of a brach point, while the
%           corresponding first column values give the y-axis
(i.e.
%           the row number) position of the branch point. The
third
%           column value is equal to +1 or to -1, indicating if
the
%           branch point is positive or negative. (The value of
a is
%           equal to the number of branch points, including
those
%           added at positions outside the aperture.)
%
%           [PHI,BPcount,phi,BPes]=BranchPointPhase(U)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%

```

```

function [PHI,BPcount,phi,BPes]=BranchPointPhase(U)

```

```

[N,M]=size(U);
if N~=M; error('Array should be square.');
```

```

%Equations 35a/b
%Atan2 is for arg() which calculates principle value form
dPx=U(:,2:end).*conj(U(:,1:end-1));
dPx=atan2(imag(dPx),real(dPx));
dPy=U(2:end,:).*conj(U(1:end-1,:));
dPy=atan2(imag(dPy),real(dPy));

BPes=[];
curl=dPx(1:end-1,:)+dPy(:,2:end)-dPx(2:end,:)-dPy(:,1:end-1); %Eqn 36
for n=1:N-1
    for m=1:M-1

```

```

        %      if abs(curl(n,m))>1e-3 %Tollerace for Branch point from
Curl of unit
        if abs(curl(n,m))>.1*pi
            BPes=[BPes; n m sign(curl(n,m))];
        end
    end
end
BPcount=size(BPes,1);
bpcount=BPcount;

Uh=ones(N,M);
if bpcount>0
    BPexcess=sum(BPes(:,3));

    %Concerned with ensuring the number of positive and negative branch
    %points are the same. And if not carry out this process
    while BPexcess~=0 %Calculating Step 2 1/2 (eqn 44)
        R=(N+3)/2;
        theta=(0:359)*pi/180;
        x=R*cos(theta)+M/2;
        y=R*sin(theta)+N/2;
        V=zeros(1,360);
        for k=1:BPcount
            V=V+BPes(k,3)./sqrt((x-BPes(k,2)).^2+(y-BPes(k,1)).^2);
        end
        [mx,ii]=max(BPexcess*V);
        bpcount=bpcount+1;
        BPes=[BPes; y(ii) x(ii) -sign(BPexcess)];
        BPexcess=sum(BPes(:,3));
    end
    [x,y]=meshgrid(1:M,1:N);
    for bpc=1:bpcount
        X=BPes(bpc,2)+0.5;
        Y=BPes(bpc,1)+0.5;
        pn=BPes(bpc,3);
        if pn>0
            Uh=Uh.*(x-X)+i*(y-Y);
        else
            Uh=Uh./[(x-X)+i*(y-Y)];
        end
    end
end

phi=angle(Uh);
ii=find(~isfinite(U));
phi(ii)=NaN;

u=U./Uh;

dpx=u(:,2:end).*conj(u(:,1:end-1));
dpx=atan2(imag(dpx),real(dpx));
dpy=u(2:end,:).*conj(u(1:end-1,:));
dpy=atan2(imag(dpy),real(dpy));

```

```

Phi=Reconstructor(dpx,dpdy);
PHI=Phi+phi;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Code provided by Dr. David Fried as apart of NVWCER algorithm fo
% Function, called Reconstructor, used to accomplish reconstruction
based on
% simple addition of phase differences. Starting from the center of the
array
% first the phase differences along the y-axis direction are added for
the
% central line. Then starting from each point in that line the phase
% differences are added along the x-axis direction.
% The process starts with the phase of the central element treated as
being
% equal to zero, but finally the phase of all elements are adjusted so
that
% mean phase is equal to zero.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%
%      INPUTS
%      pdx (N,N-1)  = Phase difference for the x-direction, i.e., for the
%                      direction in which the column numbers change (in
%                      radians).
%      pdy (N-1,N)  = Phase difference for the y-direction, i.e., for the
%                      direction in which the row numbers change (in
radians).
%
%      OUTPUTS
%      phi (N,N)     = Reconstructed phase (in radians).
%
%      phi=Reconstructor(pdx,dpdy)
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function phi=Reconstructor(pdx,dpdy)

if nargin==2
    yn=-1;
end

[a,b]=size(pdx);
if a~=b+1
    error('Size of pdx is not N-by-(N-1).')
end
[c,d]=size(dpdy);
if (d~=a) | (c~=b)
    error('Size of pdy does not properly correspond to that of pdx.')
end

```

```

N=size(pdx,1);
hN=round(N/2);
pd=-flipud(pdy(1:hN-1,hN));
phi=flipud(cumsum(pd));
pd=pdy(hN:N-1,hN);
phi=[phi; 0; cumsum(pd)];

pd=-fliplr(pdx(:,1:hN-1));
phi=fliplr(cumsum([phi pd],2));
pd=pdx(:,hN:N-1);
phi=[phi(:,1:hN-1) cumsum([phi(:,hN) pd],2)];

W=isfinite(phi);
phi=phi-mean(phi(W(:)));
phi(~W)=0;

```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

- Allen, M. R. (2007). Wavefront control for space telescope applications using adaptive optics. (Master's thesis, Naval Postgraduate School.) Retrieved from: <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA475847>
- Baker, K. L. (2005). "Least-squares wave-front reconstruction of Shack-Hartmann sensors and shearing interferometers using multigrid techniques." *Review of Scientific Instruments* 76.5: 053502. Print.
- Corley, M. S. (2010). Maritime adaptive optics beam control. (Master's thesis, Naval Postgraduate School.) Retrieved from: <http://www.dtic.mil/dtic/tr/fulltext/u2/a531558.pdf>
- Dai, G.M. (2008). *Wavefront optics for vision correction*. Bellingham, WA: SPIE, Print.
- Fried, D. L. (1998). "Branch point problem in adaptive optics." *Journal of the Optical Society of America*. 15.10: 2759. Print.
- Fried, D. L. (2001) "Adaptive optics wave function reconstruction and phase unwrapping when branch points are present." *Optics Communication* 43.72. Print.
- Fried, D. L. (2013). *Report no. tn-307: simple solve*. Unpublished manuscript.
- Mann, K. *Shack-Hartmann Wavefront Sensor*. Digital image. *Llg-ev*. N.p., n.d. Web. 13 June 2012. http://www.llg-ev.de/uploads/pics/kwl_006_01.jpg
- Pellizzari, C. C. (2007). Phase unwrapping in the presence of strong turbulence. (Master's thesis, Air Force Institute of Technology.) Retrieved from: <http://www.dtic.mil/dtic/tr/fulltext/u2/a517280.pdf>
- Primot, J. (2003). "Theoretical description of shack-hartmann wave-front sensor." *Optics Communications* 222.1-6: 81-92. Print.
- Roggemann, M. C., & Welsh, B. (1996). *Imaging through turbulence*. Boca Raton: CRC Press, Print.
- Schmidt, J. D. (2010). *Numerical simulation of optical wave propagation*. Bellingham: SPIE, Print.
- Southwell, W. H. (1980). Wave-front estimation from wave-front slope measurements. *Journal of the Optical Society of America*, 70(8), 998-1006. Print.
- Wyant, J. C. (2003). Zernike polynomials for the web [pdf]. Retrieved from: www.optics.arizona.edu/jcwyant/Zernikes/ZernikePolynomialsForTheWeb.pdf

Zhu, L., Sun, P.C., Bartsch, D. U., Freeman, W. R., & Fainman, Y. (1999, January). Adaptive control of a micromachined continuous-membrane deformable mirror for aberration compensation. *Applied Optics*, 38, 168-176. Print.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California